Coding of depth maps by elastic deformations of curves

Marco Calemme

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-032

EDITE – ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

Télécom ParisTech

Spécialité « Signal et Images »

présentée et soutenue publiquement par

Marco CALEMME

20 Septembre 2016

Codage de cartes de profondeur par deformation de courbes elastiques

Coding of depth maps by elastic deformations of curves

Directeurs de thèse :
Béatrice PESQUET-POPESCU
Marco CAGNAZZO

Jury

M. Patrick LE CALLET, Professeur, Polytech Nantes/Université de Nantes	Rapporteur
M. Laurent LUCAS, Professeur, Université de Reims-Champagne-Ardenne	Rapporteur
M^{me} Luce MORIN, Professeur, INSA Rennes	Examinateur
M. Eric MERCIER, Ingénieur de recherche, Zodiac Data Systems	Examinateur

THÈSE

Télécom ParisTech
Grande école de l'Institut Télécom - membre fondateur de ParisTech
46 rue Barrault — 75634 Paris Cedex 13 — Tél. +33 (0)1 45 81 77 77 — www.telecom-paristech.fr

Abstract

Dans le format multiple-view video plus depth, les cartes de profondeur peuvent être représentées comme des images en niveaux de gris et la séquence temporelle correspondante peut être considérée comme une séquence vidéo standard en niveaux de gris. Cependant les cartes de profondeur ont des propriétés différentes des images naturelles: ils présentent de grandes surfaces lisses séparées par des arêtes vives. On peut dire que l'information la plus importante réside dans les contours de l'objet, en conséquence une approche intéressante consiste à effectuer un codage sans perte de la carte de contour, éventuellement suivie d'un codage lossy des valeurs de profondeur par-objet.

Dans ce contexte, nous proposons une nouvelle technique pour le codage sans perte des contours de l'objet, basée sur la déformation élastique des courbes. Une évolution continue des déformations élastiques peut être modélisée entre deux courbes de référence, et une version du contour déformée élastiquement peut être envoyé au décodeur avec un coût de codage très faible et utilisé comme information latérale pour améliorer le codage sans perte du contour réel. Après que les principales discontinuités ont été capturés par la description du contour, la profondeur à l'intérieur de chaque région est assez lisse. Nous avons proposé et testé deux techniques différentes pour le codage du champ de profondeur à l'intérieur de chaque région. La première technique utilise la version adaptative à la forme de la transformation en ondelette, suivie par la version adaptative à la forme de SPIHT. La seconde technique effectue une prédiction du champ de profondeur à partir de sa version sous-échantillonnée et l'ensemble des contours codés.

Il est généralement reconnu qu'un rendu de haute qualité au récepteur pour un nouveau point de vue est possible que avec la préservation de l'information de contour, car des distorsions sur les bords lors de l'étape de codage entraînerait une dégradation évidente sur la vue synthétisée et sur la perception 3D. Nous avons étudié cette affirmation en effectuant un test d'évaluation de la qualité perçue en comparant, pour le codage des cartes de profondeur, une technique basée sur la compression d'objects et une techniques de codage vidéo hybride à blocs.

Abstract

In multiple-view video plus depth, depth maps can be represented by means of grayscale images and the corresponding temporal sequence can be thought as a standard grayscale video sequence. However depth maps have different properties from natural images: they present large areas of smooth surfaces separated by sharp edges. Arguably the most important information lies in object contours, as a consequence an interesting approach consists in performing a lossless coding of the contour map, possibly followed by a lossy coding of per-object depth values.

In this context, we propose a new technique for the lossless coding of object contours, based on the elastic deformation of curves. A continuous evolution of elastic deformations between two reference contour curves can be modelled, and an elastically deformed version of the reference contours can be sent to the decoder with an extremely small coding cost and used as side information to improve the lossless coding of the actual contour. After the main discontinuities have been captured by the contour description, the depth field inside each region is rather smooth. We proposed and tested two different techniques for the coding of the depth field inside each region. The first technique performs the shape-adaptive wavelet transform followed by the shape-adaptive version of SPIHT. The second technique performs a prediction of the depth field from its subsampled version and the set of coded contours.

It is generally recognized that a high quality view rendering at the receiver side is possible only by preserving the contour information, since distortions on edges during the encoding step would cause a sensible degradation on the synthesized view and on the 3D perception. We investigated this claim by conducting a subjective quality assessment test to compare an object-based technique and a hybrid block-based techniques for the coding of depth maps.

Keywords: multi-view video plus depth, depth map coding, contours, lossless representation, elastic curves, image synthesis, quality assessment.

Table of Contents

List of Figures

List of Tables

Introduction

Context and Motivation

The video-plus-depth representation for multi-view video sequences (MVD) consists of several views of the same scene with their associated depth information, which is the distance from the camera for every point in the view. The MVD representation allows functionalities like 3D television and free-viewpoint video [MWS06, VWS11, GPT+07], generating large volumes of data that need to be compressed for storage and transmission. As a consequence, MVD compression has attracted a huge amount of research effort in the last years, while ISO and ITU are jointly developing an MVD coding standard [ISO11]. Compression should exploit all kinds of statistical dependencies present in this format: spatial, temporal and inter-view, but also inter-component dependencies, *i.e.* between color (or texture) and depth data [MSMW07b, DPPC13].

This thesis focuses in particular on depth map compression. Depth maps are not meant to be visualized by the user, they are used instead by depth-image-based rendering software (DIBR) to synthesize novel views to provide a more immersing experience through multiview screens or free-viewpoint video. Classic techniques developed for texture images do not take into account the effects caused by the compression stage on the synthetic view. Moreover, a compression technique tailored on depth maps can exploit their peculiar properties: objects within a depth map are usually arranged along planes in different perspectives; as a consequence, there are areas of smoothly varying levels, separated by sharp edges corresponding to object boundaries. These characteristics call for an accurate encoding of contour information. Some of the approaches proposed in the literature include modelling the depth signal as piecewise polynomials, such as wedgelets and platelets [MMS+09], where the smooth parts are separated by the object contours. However, it is generally recognized that a high quality view rendering at the receiver side is possible only by preserving the contour information [GLG12], [SKN+10], since distortions on edges during the encoding step would cause a sensible degradation on the synthesized view and on the 3D perception. In other words, a lossless or quasi lossless coding of the contour is practically mandatory. Moreover, depth map segmentation is eased by the nature of the signal, and the extraction of the contours can be achieved with specifically designed segmentation techniques [ZRS+12].

Lossless representation is usually very expensive in terms of bit-rate requirements [CPPV07], so the goal of this thesis is to investigate a new paradigm that performs a lossless encoding of objects' boundaries, exploiting the temporal and inter-view redundancy. Once developed a technique to code the objects' contours, a suitable representation for the inner part has to be researched.

Contributions

An original intuition brings to the video coding domain the framework introduced by Srivastava *et al.* [SKJJ10] to model a continuous evolution of elastic deformations between two reference curves. The referred technique interpolates between shapes and makes the intermediary curves retain the global shape structure and the important local features such as corners and bends. Moreover, it is also relatively easy to compute the geodesic between the two reference curves: according to the interpretation of the elastic metric, this geodesic consists in a continuous set of deformations that transforms one curve into another with a minimum amount of stretching and bending, and independently from their absolute position, scale, rotation and parametrization. Classical applications of elastic deformations of curves are related to shape matching and shape recognition.

A depth map coding framework has been developed, consisting of:

- a lossless contour coding technique. The proposed technique is based on the elastic deformation of curves. Using the square-root velocity representation for the elements of the curve space, a continuous evolution of elastic deformations can be modelled between two reference contour curves. An elastically deformed version of the reference contours can be sent to the decoder with a reduced coding cost and used as side information to improve the lossless coding of the actual contour. Experimental results on several multiview video sequences show remarkable gains with respect to the reference techniques and to the state of the art.

- A simple coding scheme where depth data is represented by a set of contours defining the various regions, while the inner part of the objects is compressed with a transform technique. The used transform is the shape-adaptive wavelet transform, followed by the shape-adaptive version of SPIHT (set partitioning in hierarchical trees). Experimental results prove that the proposed technique, however very simple, is able to compete with refined solutions such as HEVC due to the properties of the depth signal.

- In alternative to the previous one, a novel coding scheme where depth data is represented by a set of contours defining the various regions together with a compact representation of the values inside each region. A 3D surface prediction algorithm is

then used to obtain an accurate estimation of the depth field from the coded contours and a subsampled version of the data. An ad-hoc coding strategy are used for the low resolution data and the prediction residuals. Experimental results prove how the proposed approach is able to obtain a very high coding efficiency outperforming the HEVC coder at medium-low bitrates.

It is generally recognized that a high quality view rendering at the receiver side is possible only by preserving the contour information since distortions on edges during the encoding step would cause a sensible degradation on the synthesized view and on the 3D perception. However, to the best of our knowledge, the impact of contour-preserving depth-coding on the perceived quality of synthesized images has not been conveniently studied. Therefore another contribution to this work is an investigation by means of a subjective study to better understand the limits and the potentialities of the different compression approaches (object-based and block-based) from the user's perspective.

Our results show that the contour information is indeed relevant in the synthesis step: preserving the contours and coding coarsely the rest typically leads to images that users cannot tell apart from the reference ones, even at low bit rate. Moreover, our results show that objective metrics that are commonly used to evaluate synthesized images may have a low correlation coefficient with MOS rates and are in general not consistent across several techniques and contents.

Structure of the manuscript

This manuscript is composed of four distinct parts, corresponding to just as many chapters. In the first chapter background notions on video coding are reported. Then the second chapter introduces the developed lossless contour coding technique, based on the elastic deformation of contours. The third chapter contains the description of the two proposed techniques to code the inner depth field of the objects. The fourth chapter deals with a quality assessment study performed to evaluate the developed depth map coding framework from the user's perspective. More precisely, the manuscript is organized as follows:

- In Chapter 1 some background notions related to video compression are presented, including the concept of hybrid video coder and quality evaluation through objective metrics. Then follows a detailed description of the state-of-the-art hybrid block based compression technique: the high efficiency video coder (HEVC). It will serve as basis for comparison in the experiments described in this manuscript. Finally the 3D video representation is introduced, dealing with multiple-view video plus depth and depth-image-based rendering techniques. A section dedicated to lossless compression and to the arithmetic coder closes the chapter.

- Chapter 2 describes the lossless contour coding technique based on the elastic deformation of contours. It is the basis of the developed depth coding framework

and it is described in detail for every step: the computation of the geodesic from one reference curve to the other; how to establish a correspondence between the predicted curve and the current one, and how to use the prediction as a context to improve the performance of the arithmetic coder. Experiments and comparisons with other popular contour coding techniques and with the state-of-the-art technique are detailed.

- Chapter 3 describes the two developed depth map coding technique based on the lossless contour coding technique presented in Chapter 2. A brief overview of the compression techniques specifically targeted at depth maps is first given. Then the developed techniques are detailed, along with the experiments on depth maps alone and on synthesized images, to validate the techniques in a practical case. Along with the objective metrics, some synthesized images are reported to give the reader a visual clue of the different artifacts introduced by the examined compression techniques.

- Chapter 4 reports the details of the subjective quality assessment study performed to validate from the user's perspective one of the techniques described in Chapter 3. After a brief introduction to the basics of subjective visual quality assessment, follows an overview of the coding techniques used for the preparation of the test material. Then the test design is discussed and its set-up described. Finally from the presented results comes a further validation of the proposed approach.

This manuscript ends with a summary of the proposed methods and their associated results, as well as some perspectives for future work in this field.

Chapter 1

Background Notions

Contents

In this chapter we will discuss the general concepts of video coding (Section 1.1) up to the last in date compression standard, HEVC, which will serve as basis for comparisons in the manuscript (Section 1.2). The 3D video format multiple-view video plus depth is

introduced in Section 1.3. Finally an introduction to lossless compression, with particular care to the arithmetic coder, is provided in Section 1.5

1.1 Video Coding Principles

A video is a three dimensional signal composed of a number of images, either grayscale or color, shown in rapid succession to create the illusion of movement. The images that compose a video are called *frames*, each one indicated by $I_k(m, n)$, where m and n define the spatial position and k refers to the time instant. For digital videos, as always considered in this thesis, m and n are integers. Due to the large size of *raw* (uncompressed) video signals, compression is necessary for storage and transmission applications. The statistical and psychophysical redundancy of the video signal make possible a drastic reduction of the file size while maintaining an acceptable quality [Bov00].

In particular the *statistical redundancy* represents both the spatial correlation between pixels in the same frame, and the temporal correlation between pixels in successive frames. Typically a high degree of correlation characterize the values of the pixels in the same area, and at the same time the temporal evolution of successive frames makes them highly correlated. This concept is depicted in Figure 1.1. An example of use of spatial correlation in video coding is the prediction of a frame from previous ones. Psychophysical redundancy on the other hand exploits the response of the human visual system (HVS) to the video stimulus and the fact that not all the information conveyed by the video will be perceived by a human observer. An example of use of psychophysical redundancy in video coding is the chroma subsampling: the HVS is more sensitive to variations in brightness and less to variations in color, so generally less resources are allocated to the chroma components with respect to the luminance [Bov00].

1.1.1 General notions of image and video compression

Compression techniques usually can be divided into two main groups: lossless and lossy compression techniques. Lossless compression techniques allow the reconstruction of the original data, while providing a moderate compression ratio. Lossy compression techniques, on the other hand, can achieve a high compression ratio by accepting a lower quality for the representation of the signal. It is possible to specify, in lossy compression techniques, a certain bit-rate or target file size to achieve, and settle for the quality available at that bit-rate, or to specify a certain level of acceptable quality and settle for the rate that allows that quality. Another aspect to consider for a video codec is the complexity, which in some conditions, *e.g.* on portable devices, is limited.

A very common scheme used for the compression of still images is depicted in Figure 1.2. It consists of three main parts: transformation, quantization, and entropy coding. *Transformation* concentrates most of the information of the signal in few coefficients, thus

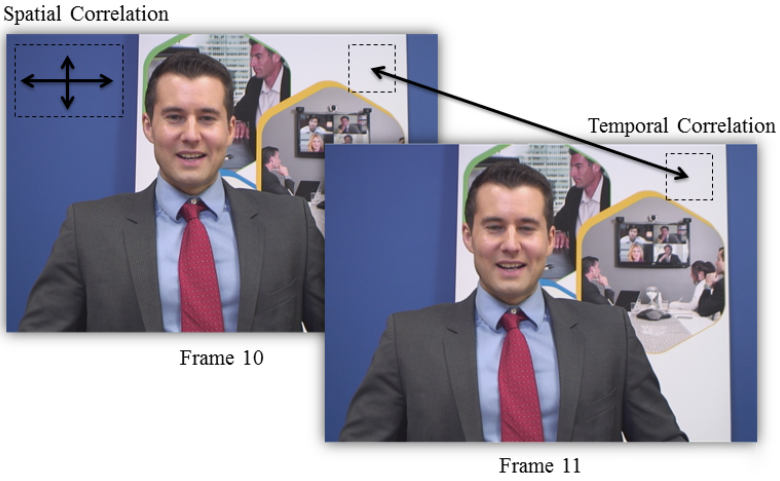


Figure 1.1: Spatial and temporal correlation.

Figure 1.2: Example of general encoder for still images.

decorrelating data; *quantization* reduces the total bit rate by using a finer representation for the most important coefficients and a coarser representation for the least important ones; finally *entropy coding* eliminates the remaining statistical redundancy.

Transformation

The goal of the transformation step is to reduce the correlation of the data [SR00], concentrating most of the information of the original signal into a small number of coefficients. The transform has to be invertible to allow the reconstruction of the signal at the decoder side. Examples of transforms used for transform coding are the Karhunen-Loeve transform (KLT), the Discrete-Fourier transform (DFT) and the Discrete-Cosine transform (DCT).

The KLT [Jai89] is the optimal transformation to decorrelate the data, however it depends on the statistics of the signal and it is built upon the assumption that the signals are stationary, which is not true for images and videos. The most used transform for images and videos is the DCT [Cla85]: it compacts efficiently the energy of the signal in few low frequency components, it is not data-dependent, and it performs better than the DFT for still images [Say12].

Quantization

After the transformation follows the quantization step, where some information is discarded by discretizing the magnitude of the coefficients. Quantization operates an association of

the input values, contained in the set S, to the discrete output subset C of cardinality N:

$$\mathrm{Q} : x \in S \rightarrow C = \{y_1, y_2, \ldots, y_N\} \ . \tag{1.1}$$

The set S is divided into N regions R_i, and their union is equivalent to the original set:

$$R_i = \{x \in S : \mathrm{Q}(x) = y_i\} \ , \ \bigcup_{i=1}^{N} R_i = S \ . \tag{1.2}$$

If for all i the amplitude of the region R_i is constant and y_i coincides with the center of the region R_i, then the quantization is said *uniform*. This solution is optimal only for uniformly distributed signals.

Another approach, called *non-uniform quantization* is to quantize coarsely the less probable value intervals of the signal, and more finely the most probable value intervals. According to the Lloyd-Max algorithm, the statistics of the signal can be used to obtain the optimal thresholds and output values for each interval, under the assumption of stationary signals [GG92].

Adaptive quantization tries to cope with this restriction, adapting the design of the quantizer to the varying statistics of the signal [Jay73][YO95]. Two classes of adaptive quantization can be distinguished: *backward adaptive quantization*, where an already transmitted portion of the signal is analysed both at the encoder and at the decoder to update the statistics used for the design of the quantizer; and *forward adaptive quantization*, that uses the statistics of a portion of the signal yet to quantize. In the case of forward adaptive quantization the statistics of the signal have to be transmitted to the decoder as a side information.

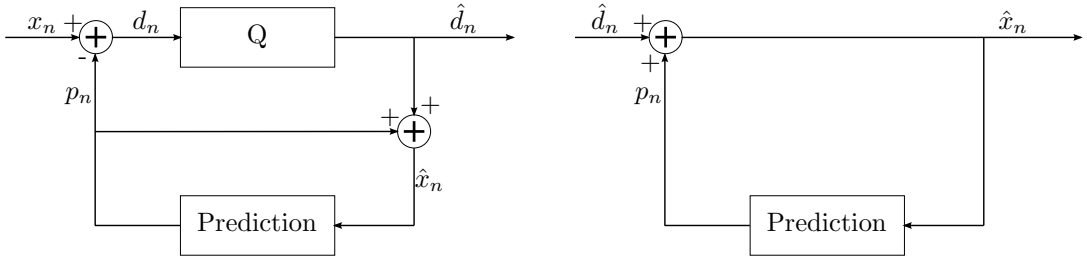
Entropic coding

The goal of the entropic coder is to remove the remaining redundancy left in the signal, reducing the average bit-rate needed to losslessly encode a sequence of symbols. This topic is discussed in detail in Section 1.5.

1.1.2 Predictive coding

As already mentioned at the beginning of Section 1.1, pixel values in video sequences have a high degree of correlation both in space and in time. Already transmitted samples of the signal can be used to make a prediction of the signal, and the difference between the signal and its prediction can be coded to increase the coding efficiency. This approach is called *differential* or *predictive coding*.

In a simple differential coding scheme (differential pulse-code modulation, DPCM) the



(a) coder (b) decoder

Figure 1.3: DPCM: coder (a) and decoder (b) scheme.

prediction is done pixel by pixel. This paradigm can be synthesized by:

$$d_n = x_n - \hat{x}_{n-1}, \tag{1.3}$$

where x_n is the actual sample, $\hat{x}_{n-1}$ is the reconstructed previous symbol, and d_n is the difference of the two. The use of the reconstructed sample makes the quantization error influence the prediction.

A general scheme for DPCM is shown in Figure 1.3. The reference for the current symbol p_n is predicted using the previous k symbols. The same prediction has to be computed both at the coder and at the decoder, so only previously transmitted symbols can be used. Considering f as prediction function, the prediction for the current symbol p_n can be defined as:

$$p_n = f(\hat{x}_{n-1}, \hat{x}_{n-2}, \ldots, \hat{x}_{n-k}). \tag{1.4}$$

A good predictor produces a difference signal with a smaller variance compared to the original one. Furthermore, a smaller distortion can be achieved if the quantization is done on the signal d_n instead of x_n.

Motion estimation and compensation

A simple differential coding scheme usually fails to exploit the temporal redundancy of video signals, for example a still object recorded by a moving camera will have a high temporal redundancy but differential coding is not able to cope with it. Motion estimation and compensation is a technique used to predict the current frame I_k from a previously transmitted frame $\hat{I}_m$ by estimating and compensating the objects' motion [Bov00] [PPCD14]. Estimated motion vectors have to be sent to the decoder in order to construct the prediction.

To perform motion estimation and compensation, the frame is first divided into macroblocks. Then the current macroblock $B_k^{\mathbf{p}}$, in position $\mathbf{p}$, is predicted by another macroblock from the frame $\hat{I}_m$, not necessarily in the same position. A *motion vector* $\mathbf{v}$ is thus defined, allowing the movement of the reference macroblock from position $\mathbf{p}$ to position $\mathbf{p} + \mathbf{v}$.

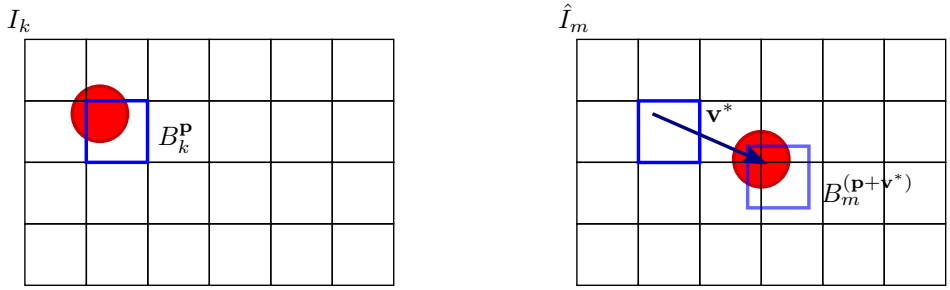


Figure 1.4: Motion estimation and compensation.

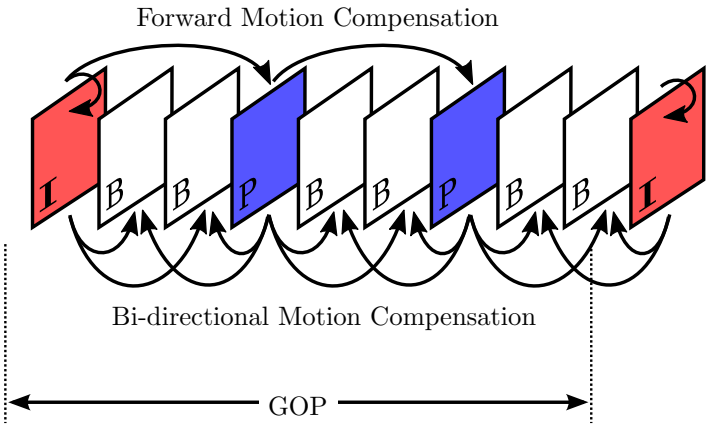


Figure 1.5: Example of general encoder for still images.

The choice of the reference macroblock is done minimizing a cost function that evaluates the similarity between two macroblocks. Given the cost function $d(\mathbf{v})$:

$$d(\mathbf{v}) = d\left(B_k^{\mathbf{p}}, B_m^{(\mathbf{p}+\mathbf{v})}\right) \ , \tag{1.5}$$

the optimal motion vector $\mathbf{v}^*$ in the search window W is:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}\in W} d(\mathbf{v}) \ . \tag{1.6}$$

An example of motion estimation and compensation is depicted in Figure 1.4.

1.1.3 Hybrid video coder

All the current video coding standards use the paradigm of the hybrid video coder. It uses two different techniques to remove the spatial and temporal redundancy, thus the name "hybrid". Transform coding is used to remove the spatial redundancy, while motion estimation and compensation to remove the temporal redundancy.

In MPEG standards the video sequence is organized in "groups of pictures" (GOPs), and the frames in a GOP can be distinguished in *intra* (or I), and *inter* frames (P and B). An example of GOP is shown in Figure 1.5. I-frames, or intra frames, are encoded

separately from other frames. Temporal redundancy is not taken into account and the transform coding paradigm allows the exploitation of only the spatial redundancy. In inter frames the temporal redundancy is exploited. More specifically for P-frames, or *predicted* frames, motion estimation and compensation is performed using as reference only previous I and P-frames of the same GOP. On the other hand for B-frames previous and following I and P-frames of the same GOP can be used as reference for motion estimation and compensation.

The generic scheme of the hybrid video coder is reported in Figure 1.6. The switch makes the system work in one of the two modes: intra-frame and inter-frame. With reference to the Figure 1.6, in *intra-frame* mode the frame I_k is first transformed by the DCT, then quantized and finally compressed with a variable length code (VLC). In *inter-frame* mode, a previously decoded frame $\tilde{I}_m$, along with motion vectors (MV) is used to predict the current frame I_k. The encoder includes then a decoder (Q^{-1} and IDCT blocks) and a buffer to store previously decoded frames, along with motion estimation and compensation procedures (ME and MC blocks, respectively). e_k is the difference between the current frame I_k and the prediction $\hat{I}_k$. The channel buffer adapts the bit-rate to the channel by controlling the quantization step. Usually with slowly changing videos the MV do not need many resources to be coded, so the quantization step can be reduced, improving the quality of the pictures. On the other hand, if the video is rapidly changing the MV will take a large amount of the available resources to be coded. To maintain the same bit-rate the quantization has to be coarser, leading to a loss of image details, but since the human visual system is unable to appreciate small spatial details in a quickly changing video the overall quality is usually not affected.

The hybrid video decoder scheme is depicted in Figure 1.7. It is simpler than the coder as it contains only the variable length decoder (VLC), inverse transform and inverse quantizer (IDCT and Q^{-1}, respectively), and the motion compensation procedure (MC). This asymmetry is convenient for all the applications where the video is encoded once and decoded many times.

1.1.4 Quality evaluation: objective metrics

The solution to the problem of the evaluation of lossy compression techniques for videos and images can be conducted through subjective or objective methods [Ric04]. In Subjective methods a group of people is asked to judge the quality of the signal. Subjective methods are generally more accurate than objective metrics, but at the same time slower and more expensive. Objective metrics use the statistical properties of the signals to assess their quality and they can be distinguished into perceptual and non-perceptual metrics.

In this section the definition of some common objective metrics is reported.

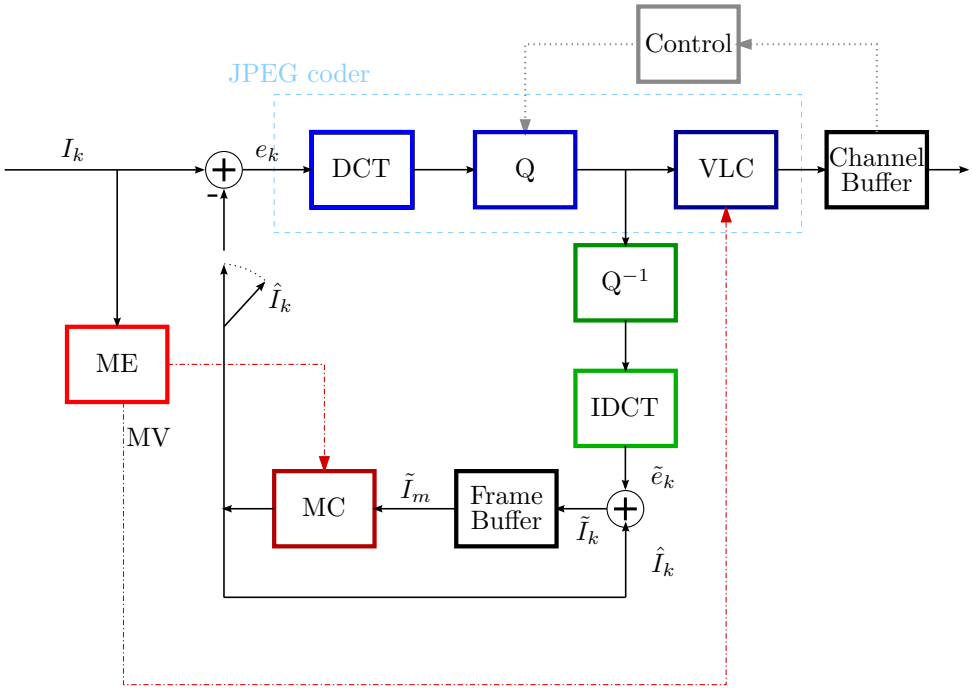


Figure 1.6: Generic hybrid video coder scheme.

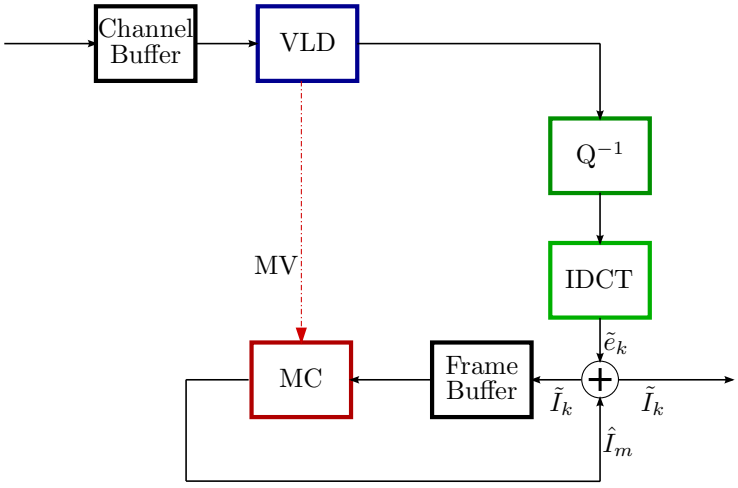


Figure 1.7: Generic hybrid video decoder scheme.

Mean absolute difference - MAD The *mean absolute difference* is a measure of statistical dispersion equal to the average absolute difference between x, the original signal, and x', the compressed one. It is defined as:

$$\mathrm{MAD} = \frac{1}{N}\sum_{i=1}^{N}|x_i - x'_i| \ , \tag{1.7}$$

where N is the number of samples.

Mean square error - MSE A more common measure in the domain of image and video compression, the *mean square error* differs from the MAD for the use of the square of the difference instead of the absolute difference. It is thus defined as:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x'_i)^2 \ , \tag{1.8}$$

where N is the number of samples.

Peak signal-to-noise ratio - PSNR Another common metric is the *peak signal-to-noise ratio*. It is related to the MSE and widely used to measure the quality of the reconstruction of lossy compression codecs. The PSNR is defined as:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \ . \tag{1.9}$$

The PSNR indicates with a gain (expressed in dB) the degree of similarity between the original signal and the reconstructed signal. It is generally calculated on the whole image, and it is generally used as an approximation of the human perception for image quality. However, relying strictly on pixel comparisons it does not take into account any level of biological factors of the human visual system [HTG08].

Structural similarity index - SSIM While MSE or PSNR estimate absolute errors, the *structural similarity index* and its *multi-scale extension (MS-SSIM)* consider image degradation as perceived change in structural information: pixels that are spatially close have strong inter-dependencies. Mean, variance and cross-correlation of images are used to compute a modified local measure of spatial correlation [WBSS04]. In addition, SSIM takes into account also some perceptual phenomena, such as luminance and contrast masking [RS08].

Although originally designed for still images, SSIM has become very popular to assess the quality of video sequences as an alternative to the PSNR: the correlation with perceptual quality is generally higher. The main critique moved to its use is that it lacks temporal assessment methods, just like the PSNR [ZZRW13].

Other objective metrics Two more objective metrics have been used in this work: Weighted Signal-to-Noise Ratio and Visual Information Fidelity, a pixel-based and a non pixel-based metric, respectively. They have been designed to assess the image quality from the user's perspective, taking into account aspects of the human visual system (HVS).

The *Weighted Signal-to-Noise Ratio* (WSNR) [MV93] uses the mean square error weighted by a contrast sensitivity function (CSF) that mimics the behaviour of the human eye: a typical band-pass filter whose peak lies at around 4 cycles per degree, with sensitivity dropping both at very low and high frequencies [KB92].

The *Visual Information Fidelity* (VIF) [SB06] uses a different approach: natural images can be modelled as the output of a stochastic source and then distorted. The mutual information between the distorted and the reference images, in the context of natural image sources, can quantify perceptual image fidelity [SBdV05].

1.2 High efficiency video coder

High Efficiency Video Coding (HEVC or H.265)is a video compression standard jointly developed by MPEG and VCEG approved in 2013. It is considered as the state-of-the-art for video coding, improving in many ways the previous H.264/MPEG-4 AVC [OSS+12], adding support for higher resolution and more notably by reducing by half the bit-rate while keeping the same level of quality.

A brief overview of the coding standard is provided in this section, as it is used for comparisons in many experiments reported in this thesis. Readers interested in more detailed explanations are invited to consult the excellent references [SOHW12][SSB14].

1.2.1 Basic structures

HEVC adopts the hybrid video coder paradigm, achieving significant gain adding more flexibility to the various procedures and structures. Efficient representation of video sequences with various resolutions relies, among other factors, on coding, prediction and transform units.

Coding unit HEVC uses different sizes for coding units, allowing the codec to be optimized for different contents. With the coding tree unit structure (CTU) the image is segmented in different *coding units* (CU, conceptually correspondent to macroblocks in previous codecs). The largest coding unit (LCU) is first chosen, then the coding units resulting from the partitioning of the LCUs are organized in a quad-tree structure [HMK+10], that provides a clean and efficient structure, a cheap representation of the tree structure, and that can make use of well-know decision algorithms. The available sizes for CU are 64×64, 32×32, 16×16, or 8×8.

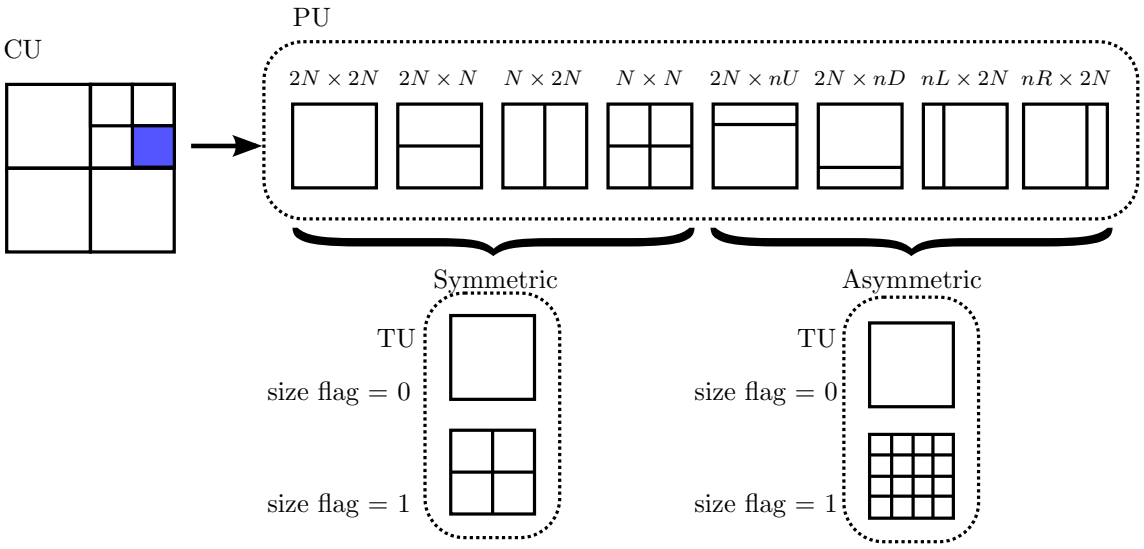


Figure 1.8: Relationship between coding units (CU), prediction units (PU) and transform units (TU).

Prediction unit The *prediction unit* is defined for each leaf node CU. It is the basic unit used for prediction and it can split the CU symmetrically or asymmetrically. In Figure 1.8 the relationship between CU and PU is shown, along with different types of splitting for PUs: a PU of size $2N \times 2N$ will have four symmetric and four asymmetric splitting structures. The asymmetric splitting is introduced to allow an efficient coding for irregular object boundaries, without this feature an oversegmentation into many small CU may have occurred.

Transform unit *Transform units*, used for the transform and quantization process, can be defined in a similar fashion to the PUs to support arbitrary shape transforms. The TUs are organized in a quad-tree hierarchy and in intra mode their size is the same or smaller than the PU, while in inter mode their size is the same as the CU and it can be smaller only if the CU has the smallest size.

The Figure 1.8 shows the relationship between CUs, PUs and TUs.

Slices and tiles In HEVC there are two structures that break processing dependencies of entire frames into smaller parts: slices and tiles.

A sequence of independently decodable CTUs is a *slice*, and a picture can be divided into any number of slices. Each slice is then divided into slice segments, where the first one contains the full slice header and the others are dependent from it. This substructure is introduced to allow low-delay transmission by sending slice segments and not full slides at the time. The subdivision of a picture into slides is illustrated in Figure 1.9(a).

Tiles are rectangular independently decodable sets of CTUs. The introduction of tiles

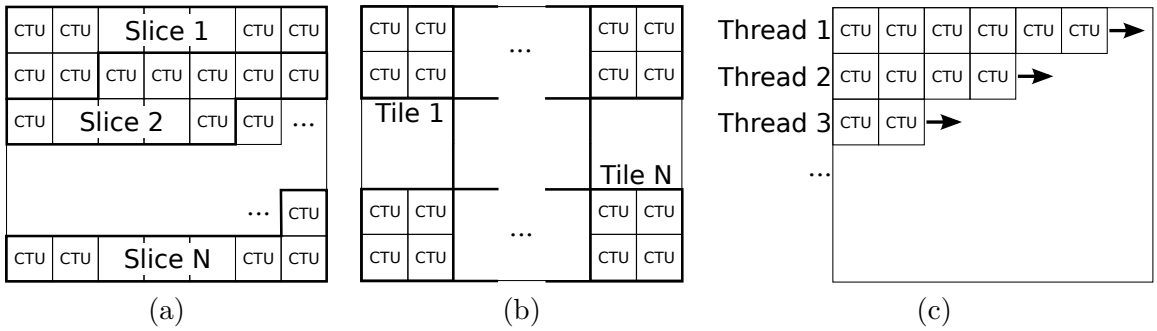


Figure 1.9: Subdivision of a picture in slices (a), and tiles (b). Wavefront parallel processing of a picture (c).

allows a more flexible classification of CTUs, a higher pixel correlation compared to slice, a higher level of parallelization and better coding efficiency as tiles do not contain header informations [SOHW12][MSH+13]. The subdivision of a picture into tiles is illustrated in Figure 1.9(b).

Another technique that allows parallel decoding is the *wavefront parallel processing* (WPP). A picture or a slice is divided into rows of CTUs, and each row can be decoded as soon as the data needed for the prediction becomes available from the previous row. The wavefront parallel processing is illustrated in Figure 1.9(c).

1.2.2 Coding

Residual coding A residual signal is coded for each CU. The supported transform sizes are 32×32, 16×16, 8×8, and 4×4. The transform used is the discrete cosine transform (DCT), with the exception of 4×4 luma residuals, where the discrete sine transform (DST) is used. The change of transform leads to the bit-rate reduction of approximately 1% for intra-predictive coding.

Transformed coefficients are then quantized, and the quantization parameter QP is used to set the quantization step q. QP can be an integer from 0 to 51, and it specifies q through the following relationship:

$$q = 2^{\frac{\mathrm{QP}-4}{6}} \ . \tag{1.10}$$

QP can be specified per sequence, frame, or CTU.

HEVC performs rate-distortion optimization (RDO) to select the best combination of transform size and intra-prediction mode.

Intra prediction Intra prediction in a certain CU follows the TU tree. The prediction is computed using the already decoded TUs that are spatial neighbours of the current one. In HEVC there are 35 different intra modes for TU spatial prediction, with size ranging from 4×4 to 32×32: 33 directional, one planar, and DC mode, as shown in Figure 1.10.

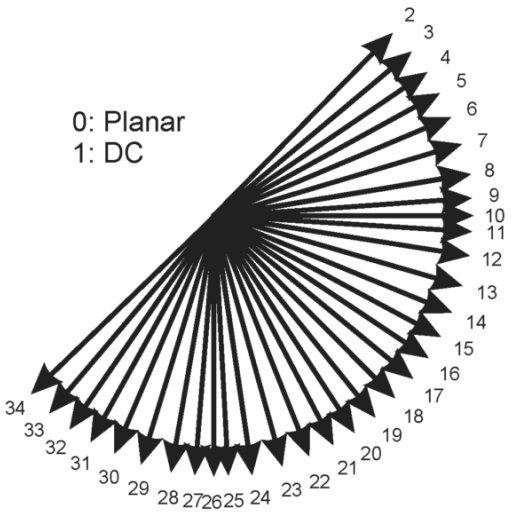


Figure 1.10: Modes and directional orientations for intra prediction.

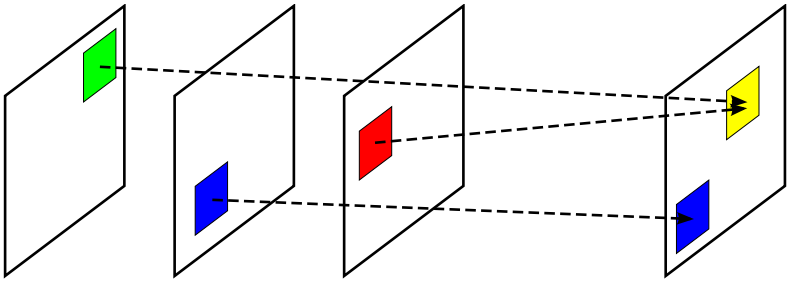


Figure 1.11: Multi-reference frame system of HEVC. PU can be taken from different reference pictures to make a prediction for the current frame.

Inter prediction: motion estimation and compensation HEVC uses candidate list indexing and it has two reference lists for motion estimation: L0 and L1. Each list can contain 16 references, however the maximum number of unique pictures is 8: a weighted prediction is performed, so an image may occur more than once in the lists. Figure 1.11 illustrates a simple example of the multi-reference frame system.

The available prediction modes are "advanced motion vector prediction" (AMVP) and "merge". In AMVP a list of candidate MVs is created, then the best candidate index is transmitted, plus the difference between the prediction and the real MV. In order to reconstruct the MV the decoder has to build the same candidate list, pick the best candidate MV, and add the difference. In merge mode no delta MV is transmitted and the candidate list is computed from neighbouring MVs; conceptually it is a sort of "SKIP" mode.

Inter-prediction uses PU samples taken from a reference picture, identified by the reference picture index, dislocated by the components of the motion vector. The resolution for the positioning of PB samples is of 1/4 the distance between the elements of the luma

and for the chroma determined by its sampling. For example, motion vectors for the chroma are specified in 1/8-pel for the format 4:2:0.

Sub-pixel precision increases the accuracy of the estimation, but it requires more memory and computational power. Consequently in the HEVC standard some limits are set: only intra-prediction is possible for 4×4 regions, while 4×8 and 8×4 regions are limited to unidirectional (forward) inter-prediction, and the smallest region that allows bidirectional prediction is 8×8.

1.2.3 In-loop filters

The HEVC standard provides two in-loop filters: deblocking and sample adaptative offset.

The *deblocking* filter is performed on the 8×8 grid, deblocking first the vertical edges, and then the horizontal edges. Since the grid is fixed and there is no filter overlap, parallelization is possible.

After the deblocking filter, the *sample adaptive offset* (SAO) is applied in the prediction loop to each CTU. This filter aims at reducing the "banding" artifacts that may appear in smooth areas by adding an adaptive offset to the decoded samples.

1.2.4 Entropy coding

Finally, HEVC performs the entropy coding using CABAC (context-adaptive binary arithmetic coder). Entropy coding is discussed in detail in Section 1.5 and with particular attention to video codecs in Section 1.5.3.

1.3 3D video representation and coding

To offer a more realistic experience than the traditional video, *3D video formats* (3DV formats) were proposed, allowing 3D display systems to support the depth perception of a visual scene. To create the illusion of depth, the display systems currently available use hi-resolution displays and parallax barriers, special-purpose glasses, or a microprism array in the case of auto-stereoscopic displays [BWS+07]. Another goal of 3DV formats is to provide *free view-point video* (FVV), a video where viewpoint and direction of the camera can be set by the user.

While stereoscopic video requires only two views, multiview displays and free view-point video require a higher number of views of the scene. Moreover, in FVV, it is possible to synthesize novel views to allow the user to navigate freely in the scene.

Depth-based representations are a class of emerging 3D video formats: a depth map, or the distance between the camera and the objects in the scene, is associated to a view to provide geometry information. In Figure 1.12 a scene represented with the video plus depth format is reported. Depth-image-based rendering techniques (DIBR) make use of the depth information to synthesize novel views, to adjust the depth perception in some

(a) (b)

Figure 1.12: Video plus depth format: texture (a) and depth map (b) from the same view point.

stereo displays, or advanced stereoscopic processing. The format that uses only one view with its associated depth map cannot handle occlusions, thus the *multiview video plus depth* (MVD) has been considered for 3D video representation. In MVD a single scene is recorded simultaneously by an array of N cameras, arranged in different spatial positions, while the depth information can be estimated from different views or obtained with special range cameras. The drawback of MVD is that the amount of data required for storage and transmission increases proportionally with N. However the redundancy of this class of signals, present both in the spatial and time domain, can be exploited for compression [VM13].

1.3.1 Depth representation

Depth maps provide a general geometric description of the scene for 3D video formats. The distance information is generally recorded as a grayscale depth image.

For synthetic sequences the geometric information is directly available, while for natural images it has to be either recorded by range cameras, that measure the time-of-flight of a laser or light pulse, or estimated from different views. Depth estimation techniques rely on the principle that the same point in two different views will be slightly displaced, and the measure of the displacement (or *disparity*) is related to the distance from the camera system. Let us consider a simple camera system with two parallel cameras, as depicted in Figure 1.13. Be P a point with distance z from the camera system, which is composed of two cameras at distance Δ_s, with focal length f. The point P will be projected on the sensors of the two cameras with a displacement with respect to the center on the right of d_1 for the first camera, and to the left of d_2 for the second camera. So the following relations can be extrapolated:

$$\frac{d_1}{x_d} = \frac{f}{z} \ , \tag{1.11}$$

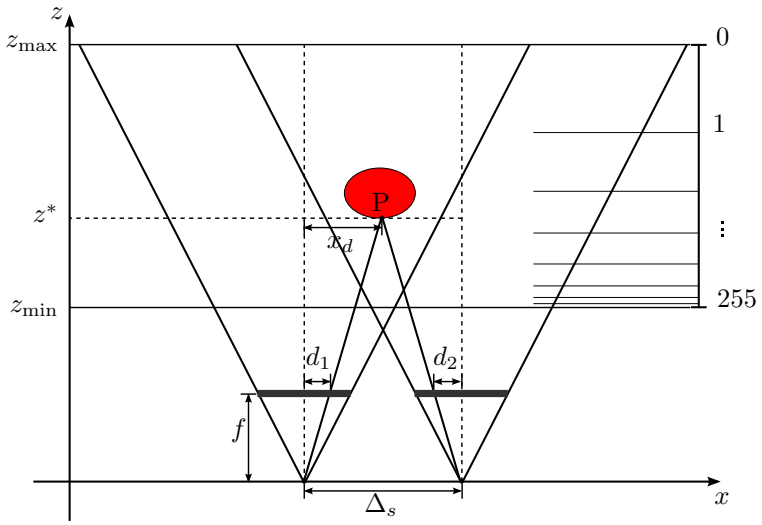


Figure 1.13: Relation between depth and disparity values.

for the camera on the left, and

$$\frac{d_2}{\Delta_s - x_d} = \frac{f}{z} \ , \tag{1.12}$$

for the camera on the right. The disparity d for each pixel is the sum of the two displacements d_1 and d_2:

$$d = d_1 + d_2 = \frac{f x_d}{z} + \frac{f(\Delta_s - x_d)}{z} = \frac{f \Delta_s}{z} \ . \tag{1.13}$$

Leading to the following expression for the relationship between d and z:

$$d = \frac{f \Delta_s}{z} \ . \tag{1.14}$$

The inverse values of z are usually quantized to operate with 8-bit resolution data, taking in account z_{min} and z_{max} for optimal usage of the depth range. The inverse depth values $I_d(z)$ are defined as:

$$I_d(z) = \text{round} \left[255 \cdot \frac{\left(\frac{1}{z} - \frac{1}{z_{\text{max}}} \right)}{\left(\frac{1}{z_{\text{min}}} - \frac{1}{z_{\text{max}}} \right)} \right] \tag{1.15}$$

If the two cameras are aligned, the disparity estimation is a 1-D problem because the displacement can only be horizontal. The existent approaches for disparity estimation can be distinguished in local and global methods. Local methods compute disparity independently for each block of the image, minimizing the cost function. On the other hand, global methods, other than minimizing the cost function per block, apply smoothness constraints to deal with the problems caused by uniform areas or changes in illumination [EG14].

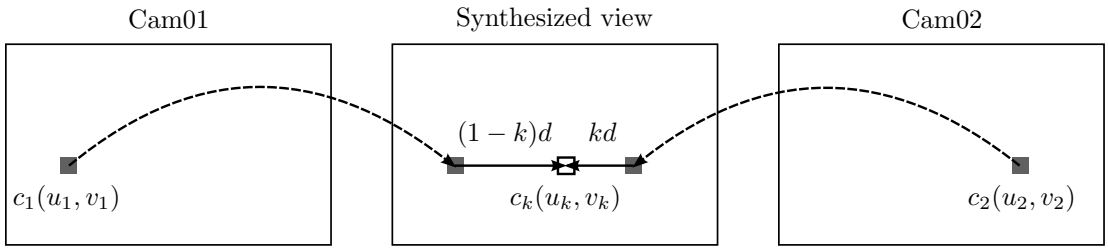


Figure 1.14: View synthesis with depth-image-based rendering (DIBR), from cam01 and cam02 to a virtual view.

1.3.2 Depth-image-based rendering

The technique called *depth-image-based rendering* uses the depth map and the texture video of one or two views to generate a synthetic view in a different point in space. The the rendering (or warping) process first projects the reference image into a 3D space, then it back-projects the scene onto the target image plane [Dar09] [McM97].

In current video solution, a strictly parallel camera scenario is enforced, and the DIBR process is simplified to a horizontal sample shift problem, from the original frame to the target frame. The shift values are calculated from the equations 1.14 and 1.15:

$$d = f\Delta s \frac{I_d(z)}{255}\left(\frac{1}{z_{\min}} - \frac{1}{z_{\max}}\right) + \frac{1}{z_{\max}} \ . \tag{1.16}$$

The focal length f, the inverse depth values $I_d(z)$, the baseline Δs and the values $z_{\min}$ and $z_{\max}$ have to be known.

In Figure 1.14 an example of the synthesis principle is shown, where a novel view is synthesized between the two cameras *cam01* and *cam02*. $k \in [0, 1]$ is the parameter that represents the relative position from *cam01* and *cam02*, c_1 and c_2 are the texture samples from the two cameras, at positions (u_1, v_1) and (u_2, v_2) respectively. The relations between original and shifted versions of c_1 and c_2 are:

$$c_{k1}(u_k, v_k) = c_1(u_1 + (1-k)d, v_1) \ , \quad c_{k2}(u_k, v_k) = c_2(u_2 + kd, v_2) \ . \tag{1.17}$$

Color blending can be applied if the two samples exhibit different color, resulting in:

$$c_k(u_k, v_k) = (1-k)c_{k1}(u_k, v_k) + kc_{k2}(u_k, v_k). \tag{1.18}$$

Using $k < 0$ or $k > 1$, one can also extrapolate a view outside the original range of the two cameras. Moreover, if one of the two samples is not available due to occlusion problems or because it is out of the view field of one of the two cameras, the synthesized texture sample $c_k(u_k, v_k)$ is obtained by shifting the available sample without color blending [MSD+08].

1.3.3 Compression

Efficient compression of the 3D video signal requires the exploitation of all the existent redundancy. To maintain a backward compatibility with existing 2D video services, a standard 2D video coding technique is used for one of the views and then additional coding tools are added for each 3D component to the 2D coding structure [VM13]. 3D coding schemes can be based on different paradigms, like:

- *inter-view prediction*, given the high degree of similarity between different views, pictures from other views are used to predict the current one [MSMW07a] [VWS11] [PMCPP13] [MJCPP13c];
- *view synthesis prediction*, DIBR techniques are used to make a synthetic prediction for the current picture [MBXV06] [YV09] [VCG+12] [PMC+16];
- *depth resampling and filtering*, a reduction of resolution for depth maps can lead to substantial rate reduction, however ad-hoc resampling and filtering techniques are then needed to achieve a good quality for synthesized depth maps [OYVH09];
- *inter-component parameter prediction*, the movement vectors for the depth map can be inherited from its associated video [WSW12] [MJPPC13] [MJCPP13a] [MJCPP13b];
- *depth modeling*, the depth signal has different properties from 2D video, such as the typical smooth areas and sharp edges, and many codecs are especially tailored for this kind of characteristics. An overview of the different techniques proposed to code depth maps is provided in Section 3.1

1.4 MPEG-4 Visual: video objects

The Visual part of the MPEG-4 standard provides technologies to view, access and manipulate *video objects* rather than pixels. This paradigm is well-suited for Internet, wireless and mobile interactive applications.

The scene in the MPEG-4 visual standard consists of one or more video objects. Each video object is characterized by its temporal evolution, shape, motion, and texture. In this context the MPEG-4 standard could benefit from the lossless contour coding technique proposed in Chapter 2. Only some particular aspects of the standard are discussed in this section, readers interested in more detailed explanations are invited to consult the excellent references [EH00] [VS01].

1.4.1 Structure

The audio-visual object is the fundamental unit of the MPEG-4 standard, and the foundation of the object-based representation. The scene is described in a hierarchical way, using the following structures:

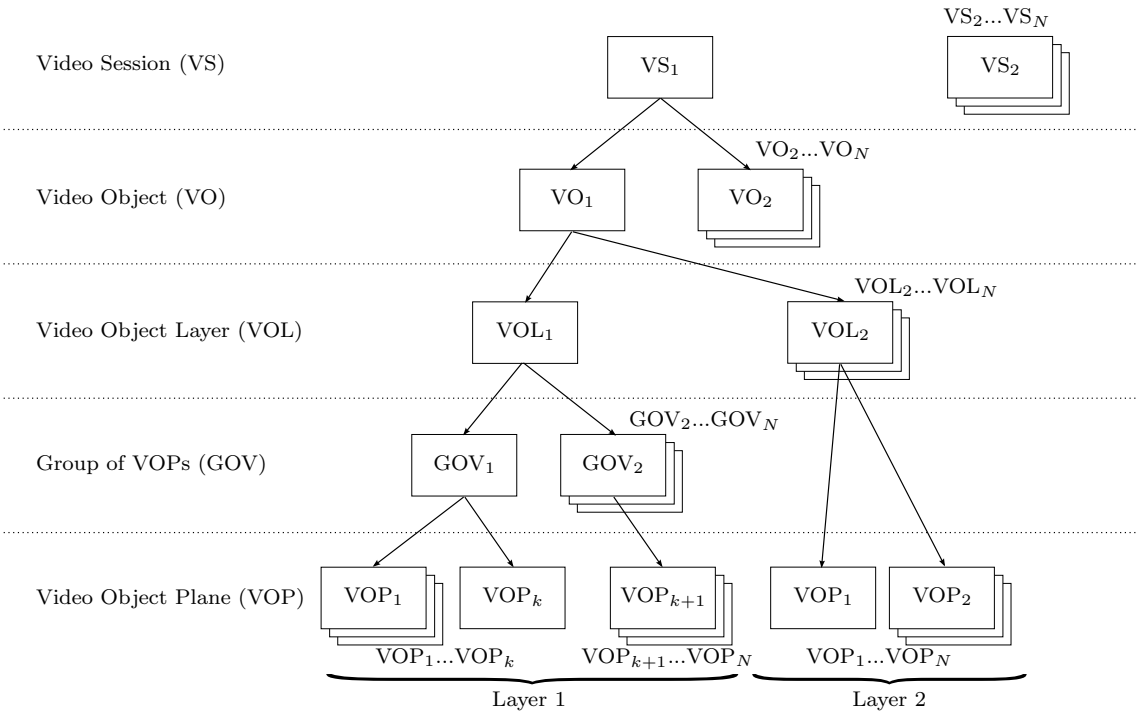


Figure 1.15: Example of MPEG-4 video structure.

- *Visual Object Sequence* (VS): the complete MPEG-4 scene which may contain any 2D, 3D, natural or synthetic object;
- *Video Object* (VO): it corresponds to a particular object in the scene;
- *Video Object Layer* (VOL): each video object can be encoded in one or more layers, to provide scalability functionalities;
- *Group of Video Object Planes* (GOV): this structure provides random access points in the bitstream, using independently encoded groups of video object planes;
- *Video Object Plane* (VOP): it is a video object at a certain time instant. VOPs can be encoded independently the one from the other, or dependently using motion estimation and compensation techniques.

The hierarchy is depicted in Figure 1.15

A VOP can contain the encoded video data of a time sample of a video object, or a *sprite* [KGS+08]. A sprite is a video object that has a wider extension than the video itself, usually it represents static or quasi static backgrounds (also referred to as "background mosaic"), to be slightly modified (warping, brightness changes) according to the framing of the video. Both video data and sprites are coded using *macroblocks*. A macroblock contains the texture information (luma and chroma, in the format 4:2:0), motion information, and shape information.

In MPEG-4 each video object is coded separately, using for VOPs a hybrid coding scheme for efficiency and backward compatibility.

1.4.2 Shape coding tools

The shape coding scheme in MPEG-4 visual relies on motion estimation to compress the shape information. Only a general description of the scheme adopted by MPEG-4 natural video standard for shape coding is in the scope of this manuscript. Interested readers are referred to [JBB+98] [KKM+98] for further information.

The shape information can be coded as binary or grayscale image. With binary shape information each pixel is labelled with 1 if it belongs to the video object at that time instant, and with 0 if it does not. The binary shape information is usually represented as an image of the same size as that of the bounding box of a VOP.

Using more values other than 0 and 1 in shape information leads to gray scale shape coding. It introduces the representation of transparent objects to reduce aliasing effects.

Binary shape coding

In MPEG-4 visual, the shape of every VOP is coded along with its texture and motion. The shape of the VOP is enclosed by a bounding box made up by blocks of size 16×16. The position of the bounding box is chosen in order to have the minimum number of blocks 16×16 with non transparent pixels. These blocks are called *binary alpha blocks* (BAB) and can be transparent if they do not contain the VOP, opaque if they are completely filled by the VOP, and "boundary" if they contain just part of the VOP. The basic tools for encoding BABs are motion compensation and the Context based Arithmetic Encoding (CAE) technique [BE97].

Grayscale shape coding

The structure for grayscale shape information is basically the same of the binary one, with the difference that every pixel can have a value from 0 to 255, according to the level of transparency of the pixel. Gray scale shape information is lossy encoded using a block-based motion compensated DCT, using binary shape coding for the coding of its support.

1.5 Lossless coding

The notion of information conveyed by a transmitted symbol is intuitively related to the degree of uncertainty of the symbol. Resting upon this observation, a measure of the information associated with an event in a probability space can be defined. Being X a discrete random variable with values $x \in \mathcal{X}$ and density function $p_X(x)$, the self-information

associated with the event $\{X = x\}$ is defined as:

$$I(x) = -\log p_X(x). \tag{1.19}$$

The self-information can be seen as a random variable $I(X) = -\log p_X(X)$, as $x \in \mathcal{X}$ varies. The mean of $I(X)$ is called *entropy* and can be seen as the measure of the uncertainty of X. The entropy is defined by:

$$H(X) = \mathbb{E}\left[I(X)\right] = -\sum_{x \in \mathcal{X}} p_X(x) \log p_X(x). \tag{1.20}$$

Using 2 as the base of the logarithm gives us a measure in bits [Sha48].

Information theory [Sha48] [CT91] shows us that the average number of bits needed to code each symbol from a stationary and memoryless source, modelled with a RV X cannot be smaller than its entropy $H(X)$. Moreover, a sequence of N *i.i.d.* random variables X can be compressed without information loss into $NH(X)$ bits, as N tends to infinity, or more bits in case N is a finite number.

Entropy coding aims at reducing the average bit-rate needed to losslessly encode a sequence of symbols, ideally up to its entropy. If a sequence is coded with fixed-length block codes, the average bit-rate would be $r = \log_2 N$. A simple approach to obtain a better performing code is to use a unique prefix code to each unique symbol that occurs in the input, where the length of the codewords is inversely proportional to the probability of the symbol, *i.e.* short codewords are used for probable symbols and long codewords for symbols with low probability. This approach is called variable-length coding (VLC). Notable examples of this strategy are Huffman coding, an optimal prefix code [Huf52], Lempel-Ziv coding [ZL77], and arithmetic coding [Sha48] [Abr63].

1.5.1 Arithmetic coding

Due to the fact that in Huffman coding the length of the codewords has to be integer, there can be up to 1 bit of information inefficiency for each codeword. Blocks of input symbols can be used to reduce this loss, but the complexity increases exponentially with the length of the blocks. Arithmetic coding uses a totally different approach to achieve a higher efficiency: the entire message is encoded as a whole into a single number n, with $n \in [0, 1[$. The compression provided by an arithmetic coder is provably optimal for an *i.i.d.* source, and near-optimal for sources that are not *i.i.d.*[CT91]. In general the first step for compression algorithms that implement arithmetic coding is to estimate a model of the probability mass function of the symbols. Source modelling also monitors the statistical properties of the data for they may vary over time. As the estimation of the symbol probabilities becomes more accurate, the output will be closer to optimal. To this purpose statistics are gathered and data contexts identified.

The arithmetic coding for a certain sequence of symbols proceeds as follows [HV92] [Sai04]:

1. the interval $[0, 1[$ is set as current interval;
2. for each symbol:
   (a) the current interval is divided into subintervals, whose number is equal to the cardinality of the input alphabet and each segment is proportional to the probability of the symbol it represents;
   (b) the subinterval that corresponds to the symbol to code is selected as new current interval;
3. the coder produces a string of bits that can distinguish the final current interval from all the others.

Some mechanism to indicate the end of the stream is needed, be it a end-of-stream symbol coded at the end, or an external indicator of the length of the sequence.

Example Let us detail a practical example: to encode the sequence of symbols [**a c** END], emitted by a source described by the (static) probabilities of its output symbols:

$$p(\mathbf{a}) = 0.6, \quad p(\mathbf{b}) = 0.2, \quad p(\mathbf{c}) = 0.1, \quad p(\mathsf{END}) = 0.1.$$

The arithmetic coding process proceeds as follows:

Current interval	Action	Subintervals **a**	**b**	**c**	END	Input
[0.000, 1.000[	subdivide	[0.000, 0.600[	[0.600, 0.800[	[0.800, 0.900[	[0.900, 1.000[	**a**
[0.000, 0.600[	subdivide	[0.000, 0.360[	[0.360, 0.480[	[0.480, 0.540[	[0.540, 0.600[	**c**
[0.480, 0.540[	subdivide	[0.480, 0.516[	[0.516, 0.528[	[0.528, 0.534[	[0.534, 0.540[	END
[0.534, 0.540[	stop					

Once reached the final interval [0.534, 0.540[, the encoder has just to output a single number that identifies it. The final interval is in binary [0.1000010110, 0.1000011100[, so the number that allows the identification of the output interval with the minimum number of bits is 0.10000110. The output of the encoder will then be **1000011**.

The decoding procedure starts by subdividing the main interval [0, 1[into 4 partitions, proportional to the symbol probabilities, then locates the input number 1000011 (equivalent in decimal to 0.538). The interval that contains the input number is again divided according to the symbol probabilities and the process is repeated until the END symbol is reached. The decoding process is outlined in Figure 1.16

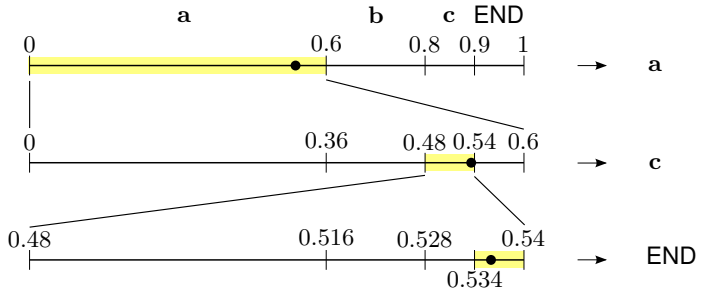


Figure 1.16: Example of arithmetic decoding. The input number is 0.538, leading to the decoding of [**a c** END].

1.5.2 Context coding

This class of lossless compression techniques do not use prior assumptions on the statistics of the data. Previously transmitted symbols provide a context to be exploited to compress efficiently the data instead [Say12]. These coding techniques analyse the history of the sequence to estimate the probabilities of the symbol to be coded.

Example This technique is particularly suited for text coding for the inherent context present in words and phrases. Let us consider an example where the the word to code is *probability*. After the coding of the first four letters *p r o b*, the next letter to code is *a*, whose probability is estimated to be around 0.06 for the English language, without any other prior knowledge. However, considering the fact that the preceding letter is *b*, the probability values for letters like *q* or *z* become very little while the probability value for the letter like *a* increases. In this example *b* is said to be the first-order context for *a*, *ob* the second-order context, and so on.

The probability of occurrence for the correct symbol increases with the length of the context, reducing the number of bits required to encode the symbol. However a high order of context is impractical, as the dictionary increases exponentially: if the input alphabet has cardinality M, the number of first-order contexts is M, for a second-order context the number increases to M^2, for a third order M^3 and so on.

A popular class of algorithms that copes with this problem is based on the *prediction with partial match* (ppm), proposed by Cleary and Witten [CW84].

1.5.3 Entropy coding in video coding standards

In the H.264/MPEG4-AVC video coder the entropy coding stage is provided by a context-adaptive variable-length code (CAVLC) for the *Baseline* and *Extended* profiles. *Main* and higher profiles use a context-adaptive binary arithmetic coder (CABAC) [MWS06]. The quite recent standard HEVC uses CABAC for all profiles [SOHW12].

In CAVLC [BL11] contexts, previously transmitted symbols, are used to improve the entropy coding performance of the coder: an adaptive estimation of the context-conditional

probability distributions allows the construction of different VLC tables, matching the conditional probabilities of the context.

At the expenses of more processing and difficulties in parallelization, CABAC [MSW03] has even better coding performances and leads to a bit-rate reduction of 10-20% [MWS06] compared to CAVLC. CABAC is composed of three components: binarization, context modeling, and binary arithmetic coding. The binarization step maps non-binary symbols into sequences of bits, which can in turn be processed by an arithmetic coder or a bypass loop.

Chapter 2

Lossless contour coding

Contents

In depth representation, arguably the most important information lies in object contours. As a consequence, an interesting approach consists in performing a lossless coding of the contour map, possibly followed by a lossy coding of per-object depth values.

In this context, a new technique for the lossless coding of object contours, based on the elastic deformation of curves [SKJJ10], can be proposed: using the square-root velocity representation for the elements of the space of curves, one can model a continuous evolution of elastic deformations between two reference contour curves. An elastically deformed version of the reference contours can be sent to the decoder with an extremely small coding cost and used as side information to improve the lossless coding of the actual contour.

Indeed, relevant results are achieved by using elastic deformation of curves to provide more effective context information to encode a curve in between the two reference ones: we use the portion of the predicted curve corresponding to the a certain point on the curve we want to code, along with already encoded samples, to estimate the most probable direction of the following point of the contour. This direction is in turn used to parameterize the probability distribution of the symbol corresponding to the next point in the curve representation. Finally, this symbol is encoded with a context-based arithmetic encoder.

Experimental results show remarkable rate reductions with respect to standards (around -65% with respect to JBIG2), to commonly used algorithms (around -20% with respect to arithmetic coding plus differential chain coding), and to the state-of-the-art method in [DCF12] (around -6.5%).

The outline of the chapter is as follows. We first recall basic notions on shape representation, elastic deformations of curves and how they are used in the context of depth map coding, as well as the basics of arithmetic edge coding using the method of [DCF12] in Section 2.1. The proposed technique is then described in Section 2.2, experimental results and conclusions are presented respectively in Section 2.3 and Section 2.4.

2.1 Background notions

2.1.1 Shape representation and coding

The definition of *"shape"* usually relies on practical, ordinary examples that consider an object or a data set and one of its properties which is invariant to translation, rotation and scaling [Sma12] [DM98]. The concept of connectivity can be used to provide a more rigorous definition: any connected set of points is a shape [CC00]. This definition is valid either in a continuous space or in a discrete one. The shapes considered in the following chapters have the property of having been spatially quantized, as they are obtained from the capture of the image of an object through a digital acquisition device (*e.g.* a camera). In the case of a digital image a discrete shape will thus correspond to a finite set of points in the orthogonal lattice.

To represent a shape one needs to choose a set of characterizing features that allow the reconstruction of the shape from such features, be it exactly or within a certain degree of precision. Two shapes are *equivalent* if all their properties are equal. This notion can be expressed using the feature vectors of the shapes, $\mathbf{F}$ If $\mathbf{F}_A = \mathbf{F}_B$ the shapes A and B are equivalent. The *similarity* of two shapes can be defined in a similar fashion, only the distance between the feature vectors has to be smaller than a certain quantity: $||\mathbf{F}_A - \mathbf{F}_B|| < \varepsilon$.

Different kind of approaches can be used to represent shapes, in particular three main groups can be distinguished: representation based on transforms, on regions, and on

contours.

Transform-based techniques are not only used for shape representation, as in the transformed domain it can be easier to obtain descriptors for the shape. They can be linear transforms (*e.g.* Fourier, Karhunen-Loève, cosine, wavelet, etc.) or non-linear (*e.g.* Time-frequency distributions, mathematical morphology, etc.)

Region-based techniques can be distinguished into region-decomposition techniques, bounding regions, and extraction of internal features. In region-decomposition techniques the shape is decomposed in simpler structures, such as polygons, and represented with a combination of them. Using a bounding region is another possibility to represent a shape: an approximation is provided using a bounding box, a Feret box, or a convex hull. As for the internal features, the shape is represented by a set of features extracted from its internal region. Examples of this category are the skeleton transform and distance transform.

Contour-based techniques will be discussed in detail, as they will be used extensively in Chapter 2. Three classes can be distinguished:

- *set of contour points:* the shape is represented simply by a set of point, with no order relationship between them;
- *parametric contours:* a parametric representation can be attained by adding a sequential order to a set of points;
- *curve approximation:* the shape outline is decomposed into a set of geometric primitives, like straight line segments (polygon approximation) or splines [O'C97] [KKM+98] [KBE00].

For a parametric representation a simple vector of values $(x(t), y(t))$ can be used, where x and y are the coordinates of the contour points and t is the parameter. An equivalent complex representation can be defined as:

$$u(t) = x(t) + \jmath y(t), \tag{2.1}$$

where $u(t)$ is the contour signal, $x(t)$ is its real part, $y(t)$ its imaginary part, and $\jmath$ the imaginary unit.

2.1.2 Chain-coding and differential chain-coding

Chain-coding is the most common method to losslessly encode boundary pixels. A chain-code follows the contour of an object and encodes the direction of the next boundary pixel with respect to the current one [Fre61]. Since an object tends to have a quite regular contour it is usually more convenient to code the difference of direction with respect to the previous one, thus leading to *differential chain-codes* [Fre78]. An example of differential chain-coding of the 8-connected contour of an object is given in Fig. 2.1: using the dotted pixel as starting point, the contour of the object can be coded as: (3), 2, -1, 0, 3, -1, -1, 3,

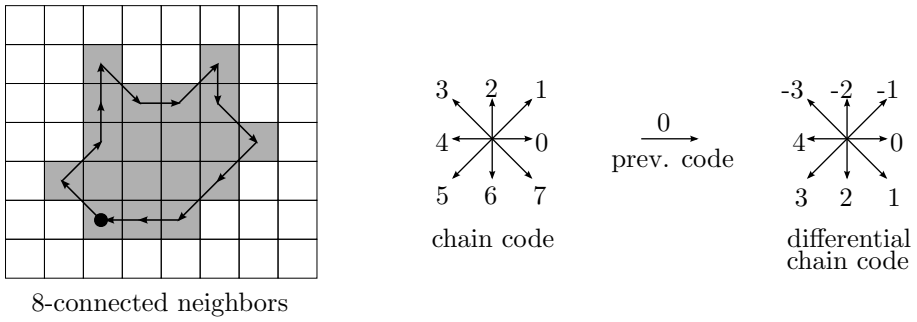


Figure 2.1: Object contour, symbols of chain code with respect to a central pixel and the differential chain-code if the previous symbol was a "0".

-1, 2, 0, 1, 0. The first symbol, indicated in parenthesis, has to be coded separately from the others and it belongs to the set of chain-code symbols.

To code a shape with a (differential) chain-coding technique the sequence of symbols is fed into an entropic encoder, such as a variable length encoder or an arithmetic encoder, possibly using contexts to improve its performance [CAB10].

2.1.3 Elastic deformation of curves

Srivastava *et al.* [SKJJ10] introduced a framework to model a continuous evolution of elastic deformations between two reference curves. The referred technique interpolates between shapes and makes the intermediary curves retain the global shape structure and the important local features such as corners and bends.

In order to achieve this behavior, a variable speed parametrization is used, specifically square-root velocity (SRV), so that it is possible to bend one shape into another as well as stretch or compress a certain part of it. Let us introduce some notation. The curve defining the shape is denoted by p, and $t \in [0, 1]$ is the curve parameter, leading to:

$$p : [0, 1] \rightarrow (x, y) \in \mathbb{R}^2, \tag{2.2}$$

where (x, y) are the coordinates of each point in the contour. Then, p is represented in the SRV space by q:

$$q : [0, 1] \rightarrow (x, y) \in \mathbb{R}^2, \tag{2.3}$$

$$q(t) = \frac{\dot{p}}{\sqrt{||\dot{p}||}}, \tag{2.4}$$

where $||\cdot||$ is the Euclidean norm in $\mathbb{R}^2$ and $\dot{p} = \frac{dp}{dt}$. This transformation is reversible (up to a translation): the curve p can reversely be obtained from q by:

$$p(t) = \int_0^t q(s)||q(s)||ds. \tag{2.5}$$

Introducing the SRV representation is very interesting because it can be shown that the

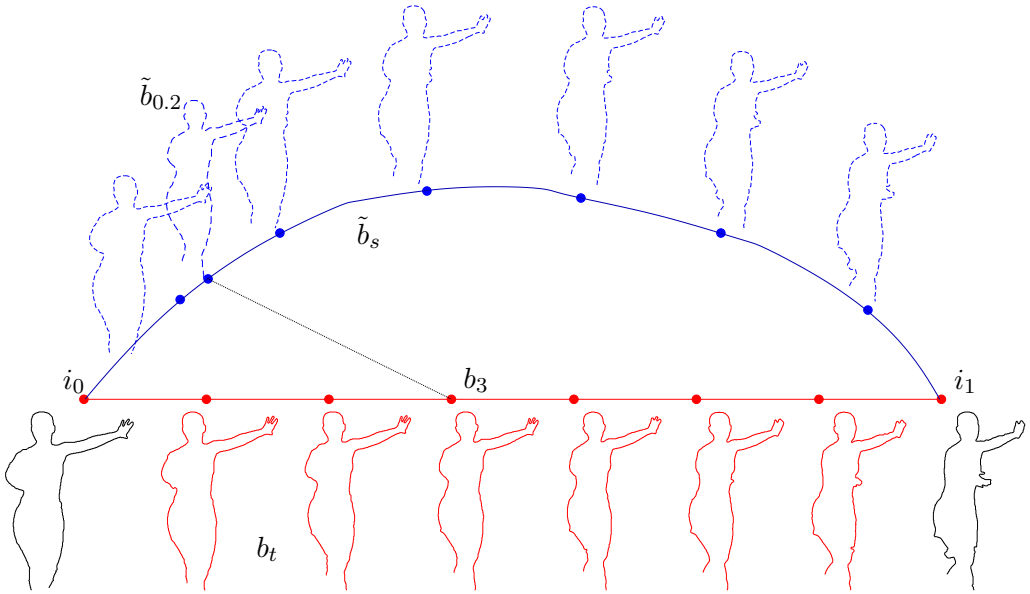


Figure 2.2: Geodesic path of elastic deformations $\tilde{b}_s$ from the curve i_0 to i_1 (in dashed blue lines). b_3 is one of the contours b_t extracted from the intermediate frames between the two reference ones, a good matching elastic curve $\tilde{b}_{0.2}$ along the path is highlighted.

simple $\mathcal{L}^2$ metric in this space corresponds to an "elastic" metric for the original curve space [MSJ07] *i.e.* a metric that measures the amount of "stretching" and "bending" between two curves, independently from a translation, scale, rotation and parametrization. Moreover, using the SRV it is also relatively easy to compute the geodesic between the two curves: according to the interpretation of the elastic metric, this geodesic consists in a continuous set of deformations that transforms one curve into another with a minimum amount of stretching and bending, and independently from their absolute position, scale, rotation and parametrization [SKJJ10].

An example of the geodesic connecting two curves is shown in Fig. 2.2. We show in black two contours extracted from the depth of the video sequence "ballet" using the Canny edge detector. These depths correspond to the views 1 and 8, at time instant 2. In red we show the extracted contours of intermediary views, while in dashed blue we show a sampling of the elastic geodesic computed between the two extreme curves: it is evident that the elastic deformations along the geodesic reproduce very well the deformations related to a change of viewpoint. Similar results where obtained in the temporal domain: given the initial and final shapes, elastic deformations are able to represent the temporal deformation of the contour of an object, at least in the case where there is a small temporal gap between the initial and final instants.

These observation lead us to conceive a lossless coding technique for object contours: supposing that the encoder and the decoder share a representation of the initial and final shape, they can reproduce the exactly same geodesic path between them. Then, the decoder will conveniently decide which point of the geodesic (one of the blue curves in Fig. 2.2)

shall be used as context information to encode an intermediary contour (one of the red curve in the same figure). The encoder will only have to send a value $s^\star \in [0, 1]$ to identify this curve. If this curve is actually similar to the one to be encoded, it is possible to exploit this information to improve the lossless coding of the latter. However the use of elastic deformation of curves for contour coding is not as easy as it appears, and the solution to this problem is one of the main contribution of this thesis.

We observe that this use of the elastic deformation tool is quite different from what it was proposed in the past. Classical applications of elastic deformations of curves are related to shape matching and shape recognition, and the only previous paper using elastic curves in compression is [AEDF$^+$15]. However, in that paper, elastic curves are used in the context of distributed video coding, in order to improve the motion-compensated fusion of background and foreground objects, while here it is used for the lossless coding of contours.

2.1.4 Arithmetic edge coding

One of the most recent and best-performing technique for lossless coding of edges was introduced by Daribo *et al.* in [DCF12], and even though it is conceived in a block-based coding environment, we retain it as a base to develop a part of our contour-based coding technique. Their main idea is that contours of a physical object possess geometrical structures that can be exploited to predict the direction of the next symbol, given a window of consecutive previous samples. A chain code is used to represent a contour, and each symbol is encoded with an arithmetic encoder that uses a probability distribution adaptively computed using the previous symbols. The probability distribution is centred around the estimated most probable direction, which in turn is obtained using linear regression on a suitable window of previous samples: the underlying assumption is that contours exhibit linear trends. The prediction of directions and the assignment of the probability values can be reproduced at the decoder, provided that some parameters are transmitted as side information.

For curves that represent object boundaries in MVD this method has much better performance than other popular techniques [DCF12], however as we noticed in our study further improvements are possible. On the one hand the linear regression can be replaced with a more effective estimator for the direction of the next symbol. On the other hand the encoding algorithm does not take advantage from temporal correlation of contours and the prediction of the most probable direction cannot cope with sudden changes of direction.

Our proposed method uses temporal correlation of contours to produce an elastic estimation of the current curve; this curve is subsequently employed to improve the statistical model of each symbol of the chain code, resulting in a remarkable improvement of the coding rate. The two reference curve used to produce the elastic deformation are encoded using a modified version of the arithmetic edge coding technique described by Daribo *et al.* [DCF12], where the linear regression is substituted with the angle of the

average direction of the symbols in the considered window of samples.

2.2 Proposed technique

We propose a technique to encode the contour of an object in a single view video sequence, in a multiview set of images or in a multiview video sequence, and in any context where two reference curves are available. The targeted application is depth coding in MVD applications, and this for two reasons: first, contour information is extremely important for depth, and its lossless representation is necessary for obtaining a good subjective quality for synthesized views; second, extracting contours from a depth map is relatively easy, since they are typically made up of smooth regions separated by sharp discontinuities. However, in this thesis we do not investigate the contour extraction from depth images, but uniquely their lossless coding; moreover, we point out that our method can also be applied on contours extracted from natural videos, as we show in the experimental results. In this last case, contours are typically extracted from precise segmentation maps: our method efficiently encodes both contours automatically extracted from depths and contours obtained by segmentation.

As already mentioned, our method apply to the case where we have a set of contours related to the same object, be them representing the temporal evolution of the object borders or their deformation related to the change of viewpoint: the example in Fig. 2.2 refers to the latter case. However, without loosing generality, we consider the following use case. We have a set of $K \geq 3$ contours ($K = 8$ in Fig. 2.2). We refer to the parametric representation of the first contour as $i_0[n] = (x_{i_0}[n], y_{i_0}[n])$, with $n \in \{1, 2, \ldots, N_{i_0}\}$; likewise, $i_1[n]$ with $n \in \{1, 2, \ldots, N_{i_1}\}$ is the parametric representation of the last contour, and $b_t[n]$ with $n \in \{1, 2, \ldots, N_t\}$ the one of the generic t-th intermediate curve, with $t \in \{1, \ldots, K-1\}$. We propose an "intra method" (i.e. without temporal or inter-view prediction) to encode i_0 and i_1, and a "bidirectional method" (i.e. with prediction from two already encoded curves) for the curves b_t. We will refer to the intra-coded contours as "I-contours" and to the $K-2$ intermediate contours, to which our prediction based method is aimed, as "B-contours".

I-contours

Indeed, the I-contours are supposed to be available at the encoder and at the decoder before the B-contours in order to develop a prediction method to code the latter. To encode the I-contours i_0 and i_1 we propose a small yet effective modification to the arithmetic edge coding technique described in [DCF12]. In particular, we consider a window of previously transmitted points, and we use them to estimate a probability distribution for the symbol that represents the next point in the curve. The predictor for the most probable direction of the next symbol is the angle of the average direction of the symbols in the

window of samples, described in detail in Section 2.2.2. An outline of the coder is given in Section 2.2.3.

B-contours

Let us now describe the proposed method for the B-contours. The basic idea is independent from the value of K and from the structure of dependencies (or, borrowing the terms from the classical hybrid video coding paradigm, from the GOP structure). More precisely, in order to encode b_t, we consider the geodesic path between i_0 and i_1. The elastic deformation tool allows us to easily generate any intermediate curve on the geodesic (dashed blue curves in Fig. 2.2), simply by specifying a position parameter $s \in [0, 1]$. Let us refer to $\widetilde{b}_s[n]$ the parametric representation of a curve on the geodesic. We observe that $\widetilde{b}_0 = i_0$ and $\widetilde{b}_1 = i_1$. Since i_0 and i_1 are available at the encoder and the decoder, they both can produce the same curve $\widetilde{b}_s$, for any s, and use it as side information for the lossless encoding of b_t. Intuition suggests that the encoder and the decoder could agree in using $s = \frac{t}{K-1}$ as the position on the geodesic used to predict b_t. However, as we will show in the experimental section, a significant coding gain can be obtained if we let the encoder select a suitable value for s, let it be $s^\star$, and use it for the encoding. Of course, $s^\star$ should be transmitted to the decoder using a suitable number of bits. The choice of $s^\star$ and the number of coding bits will be discussed in the experimental section. Likewise, the problem of the optimal "GOP structure" will be discussed in Section 2.3.4. In the rest of the current section, in order to simplify the discussion, we will consider a simple IBIBIB... configuration. This is equivalent to having $K = 3$: in other words, we encode an object contour knowing the previous and the future curves. The proposed method can however be applied to any value of K without major modifications. Other coding structures are shown in the experimental part at Section 2.3.4.

In the rest of this section we will explain how to select a suitable part of the elastic curve $\widetilde{b}_s$ to be used as context to encode the current edge symbol; how to use this information to determine the most probable direction for the next symbol on the contour; and which side information needs to be sent to the decoder such that it can replicate the same behaviour as the encoder.

2.2.1 Correspondence function

In this section we consider the encoding of a single curve b_t using the elastic representation $\widetilde{b}_{s^\star}$, with $s^\star$ suitably selected by the encoder. The curves are sampled respectively on N_b and $N_{\widetilde{b}}$ points. To simplify the notation, we will drop the subscript t and $s^\star$ where this does not give rise to ambiguities.

To use the suitable portion of the synthetic curve $\widetilde{b}$ as side information to code the current point on b, it is essential to have a function that associates each point of b to the corresponding point of $\widetilde{b}$. This function is generated at the encoder and has to be

transmitted to the decoder. In order to establish this association between the two curves, we use Dynamic Time Warping [MÖ07] (DTW). First, it is needed to establish a feature space $\mathcal{F}$ for the curves: a distance in this space will then be used to create the DTW function. Since we want to link the parts of the curve that have the same characteristics in terms of lobes, spikes and such, we tried several features such as curvature, direction of the tangent vector, and its first order derivative. The direction of the tangent vector proved to be precise and reliable and was selected for the use with DTW.

Let us use a complex representation that associates to every point (x, y) the complex number $p = x + \jmath y$, and let us refer to the sequence of directions of the tangent vector of the curve $c = (x_c, y_c)$ as φ_c:

$$\forall n \in \{1, \cdots, N_c\},\ \varphi_c[n] = \arg\left(p_c[n] - p_c[n-1]\right) \in \mathcal{F}, \tag{2.6}$$

where $\mathcal{F} = [-\pi, \pi[$. Computing Eq. (2.6) for b and $\widetilde{b}$ we obtain the sequences of features φ_b and $\varphi_{\widetilde{b}}$ defined on N_b and $N_{\widetilde{b}}$ points respectively.

A typical behavior of two feature sequences is shown in Fig. 2.3; while the two sequences have similar shapes, they are not aligned. To perform the alignment we need a local distance measure (or local cost measure), defined as $d : \mathcal{F} \times \mathcal{F} \to \mathbb{R}^+$. The local distance measure $d(\varphi_b[n], \varphi_{\widetilde{b}}[m])$ should be small when the two features are similar, large otherwise. Since in our case $\mathcal{F} \subset \mathbb{R}$, we can use as distance the square of the direction difference: $\left(|\varphi_b - \varphi_{\widetilde{b}}| \bmod \pi\right)^2$. By evaluating the local cost measure for every couple of elements in the sequences, we obtain the cost matrix $C \in \mathbb{R}^{N \times M}$, where the generic element is defined as follows:

$$C(n, m) = d(\varphi_b[n], \varphi_{\widetilde{b}}[m]) = \left(\left|\varphi_b[n] - \varphi_{\widetilde{b}}[m]\right| \mod \pi\right)^2.$$

We have to find now the sequence ψ, defined as a sequence of couples:

$$\psi[\ell] = (\nu^\star_\ell, \mu^\star_\ell) \in \{1, \cdots, N\} \times \{1, \cdots, M\},$$

such that:

$$(\nu^\star, \mu^\star) = \arg\min_{(\nu,\mu)} \sum_{l=1}^{L} C\left(\nu[\ell], \mu[\ell]\right)$$

under the conditions:

- boundary condition, $\psi[1] = (1, 1)$ and $\psi[L] = (N, M)$;
- monotonicity of $\nu^\star[\ell]$ and $\mu^\star[\ell]$;
- step size, $\psi[\ell] - \psi[\ell - 1] \in \{(0, 1), (1, 0), (1, 1)\}$.

In practice ψ is a sequence of indices $(\nu^\star[\ell], \mu^\star[\ell])$ of the curves φ_b and $\varphi_{\widetilde{b}}$, such that $\varphi_b[\nu^\star[\ell]]$ and $\varphi_{\widetilde{b}}[\mu^\star[\ell]]$ are best matched under the aforementioned constraints. The association by DTW of the two sequences is shown in dotted black lines in Fig. 2.3.

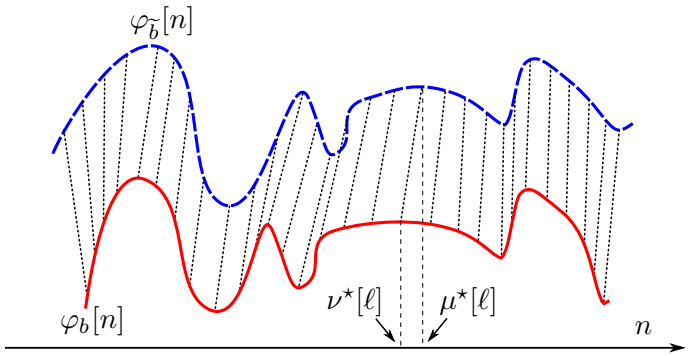


Figure 2.3: Example of association of two sequences by Dynamic Time Warping.

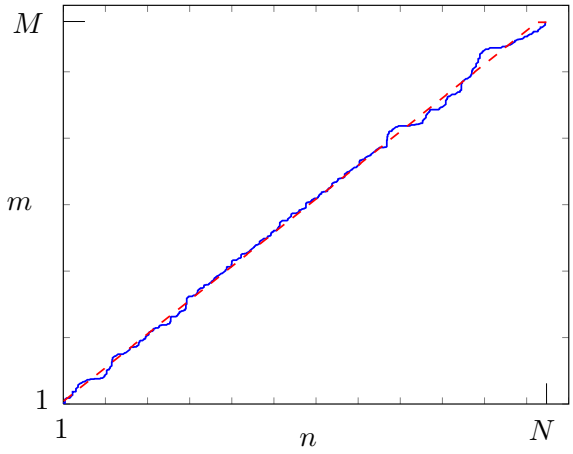


Figure 2.4: *Ballet*: correspondence function. In blue the association of the two curves using the DTW of the direction of the tangent vector, in dashed red the approximation with a first order polynomial. While n and m are the indices of samples on the curves b and $\widetilde{b}$, respectively.

In our application the correspondence function obtained with DTW on the direction of the tangent vector is very close to being affine and it can thus be approximated with a first order polynomial. The approximation has two main effects: first it reduces the number of bits needed to code the function; moreover it prevents sudden variations on the correspondence function that are rather related to outliers than actual values. However, one may wonder whether it is worth computing the exact DTW only for approximating at a first order: maybe a simple rescaling of the "temporal" axis from N points to M points could be as effective as the approximated DTW, without needing to compute the correspondence function. We have dealt with this issue with a simple heuristic approach: we compared the coding rate of our algorithm in two cases: in the first, we use the first order approximation of the DTW function; in the second we use a rescaling of N/M. We observed that using the DTW gives an average rate reduction of 5.33%. For this reason, we kept the DWT in our system.

In Fig. 2.4 there is an example of DTW and its linear approximation. The resulting correspondence between the points of the two curves is shown in Fig. 2.5. We see that

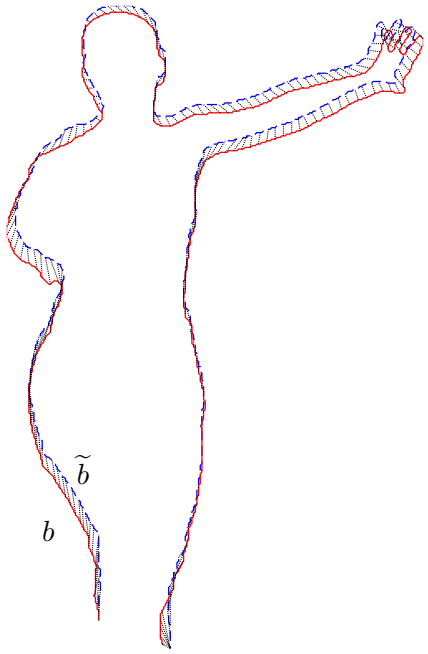


Figure 2.5: *Ballet*: correspondences between the elastic curve $\widetilde{b}$ (dashed blue) and the curve to code b (red).

all the main features of the curves are located and put in correspondence, so that for each point of the actual curve b there is an associated point on the elastic curve $\widetilde{b}$, whose neighborhood is the side information we want to use to enhance the coding of b.

2.2.2 Context

The correspondence function allows to associate the current point to a portion of the elastic curve, centered in the corresponding point. This information is used as side information to have more accurate probability values for the next symbol. We called this information the *context*, and for each point of the curve b, it is composed by:

- $\mathbf{v}_0$, a vector of N_0 points of the curve b transmitted so far (in red in Fig. 2.6);
- $\mathbf{v}_{1p}$: the "past" on the elastic curve (in dashed blue in Fig. 2.6), a vector of N_p points of $\widetilde{b}$ corresponding to $\mathbf{v}_0$; more precisely, $\mathbf{v}_{1p}$ is constituted of the points between those corresponding to the terminal points of $\mathbf{v}_0$;
- $\mathbf{v}_{1f}$: the "future" on the elastic curve, a vector of N_f points of the elastic curve $\widetilde{b}$ following the current correspondent point on $\widetilde{b}$ (in dark blue in Fig. 2.6).

Of course $\mathbf{v}_{1p}$ and $\mathbf{v}_{1f}$ are only available for B-contours, for I-contours only $\mathbf{v}_0$ is available. We use the context to obtain the most probable direction for the next symbol, then we use this result to define a distribution using the von Mises statistical model [MJ99].

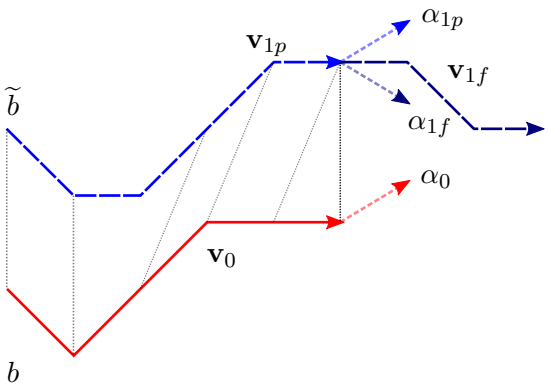


Figure 2.6: Extracts from the curves b (red) and $\widetilde{b}$ (dashed blue). The correspondences between the two curves are indicated with thin dotted black lines. The dashed lines represent the extracted direction for the vectors of points $\mathbf{v}_0$, $\mathbf{v}_{1p}$ and $\mathbf{v}_{1f}$.

Direction extraction

For all the set of points of the different curves we have to estimate the direction of the next symbol. In the case of the I-contours only the set of points $\mathbf{p}_0$ are used, while in the case of B-contours all the three sets of points $\mathbf{v}_0$, $\mathbf{v}_{1p}$ and $\mathbf{v}_{1f}$ are used.

Several approaches can be used to extract a direction from a set of ordered points. E.g. in [DCF12] a linear regression on $\mathbf{v}_0$ is used. We found that the angle of the average direction is even more effective. Using the same complex representation as in 2.2.1, the estimated direction α can be obtained using the following formula:

$$\alpha(\mathbf{v}) = \arg\left(\frac{1}{N-1}\sum_{n=2}^{N}(\mathbf{v}[n]-\mathbf{v}[n-1])\right) \in [-\pi,\pi[, \tag{2.7}$$

where $\mathbf{v}$ is the complex representation of a generic vector of N points, and is defined as $\mathbf{v} = [x_1 + jy_1, x_2 + jy_2, \cdots, x_N + jy_N]$; arg is the argument of a complex number.

Most probable direction

Applying Eq. 2.7 on the complex representation of $\mathbf{v}_0$, we obtain α_0, the angle of the average direction based solely on the previously transmitted samples. Likewise, α_{1p} and α_{1f} are the average directions based on the vectors $\mathbf{v}_{1p}$ and $\mathbf{v}_{1f}$ of the curve $\widetilde{b}$. We develop a method to estimate θ, the most likely direction of next symbol of b: we use α_{1f} to adjust the direction α_0. In this way, we manage to seize the sudden changes and go along the long trends.

A simple and intuitive formula to take into account the context for the prediction of the most probable direction is:

$$\theta = (1-q)\alpha_0 + q\alpha_{1f}, \tag{2.8}$$

with $q \in [0,1]$. This way we can weight the directions of the past of the curve b and the future of the curve $\widetilde{b}$. To decide the weight q, we observe that the side information of the elastic curve is not essential when the curves are regular, while it becomes fundamental next to the occurrence of a sudden change. So q should be small if for the current point the directions extracted from the two curves are similar (so that θ is close to α_0), and close to 1 if they are not (so that θ is close to α_{1f}):

$$q = \frac{\max\{|\alpha_0 - \alpha_{1p}|, |\alpha_0 - \alpha_{1f}|\}}{\pi}. \tag{2.9}$$

The value of q is related to the modulus of the difference of directions, and it is large if α_0 is very different from α_{1p} or α_{1f} because a dissimilarity in the neighborhood of the current point suggests a change which is not predicted solely from α_0.

Adaptive statistical model

We retain the statistical model described in [DCF12] to assign values to the symbols for the next edge. It is based on the von Mises distribution and the distribution parameters are set according to the information extracted from the curves b and $\widetilde{b}$. The von Mises distribution is the Gaussian distribution for angular measurements and it is defined as [MJ99]:

$$p(\beta|\mu,\kappa) = \frac{e^{\kappa\cos(\beta-\mu)}}{2\pi I_0(\kappa)}, \tag{2.10}$$

where $I_0(\cdot)$ is the modified Bessel function of order 0, μ is the mean and in this case it coincides with the estimated direction θ, $1/\kappa$ is the variance of the distribution. As in reference [DCF12], we set κ as a function of the predicted direction θ: when θ is aligned with the axis of the pixel connection grid, intuition tells us that it is more convenient to give a higher probability value to the symbol that represents that direction. So κ is set to:

$$\kappa = \rho\cos(2\hat{\theta}), \tag{2.11}$$

where $\hat{\theta} = \min\{|\theta - \gamma_i|\}$, and γ_i are the angles of the pixel connection grid ($\left\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \cdots\right\}$ in the case of an 8-connected grid). The parameter ρ represents a "confidence level" of the prediction: as it grows it makes the distribution more unbalanced, so the more precise is the prediction based on the context, the larger it should be to achieve higher coding gains.

2.2.3 Coding

The curve b, represented with a differential chain code, is encoded with a context-based arithmetic coder [Sai04] which for each symbol uses the probability vector assigned by the adaptive statistical model. The encoder needs to transmit to the decoder the parameters involved in the coding process:

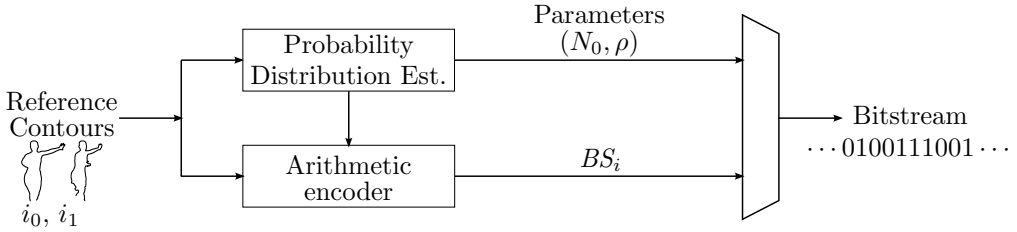


Figure 2.7: Synthetic scheme of the coder for the lossless contour coding technique for I-contours.

- $s^\star$, the selected point on the geodesic path;
- the correspondence function, approximated by a first order polynomial, so two parameters;
- the parameter ρ;
- N_0 and N_f.

With this information the decoder can reproduce the behavior of the encoder and it will compute the same probability values for each point of the curve b. We observe that N_p does not need to be sent, since it is deduced by applying the correspondence function to $\mathbf{v}_0$.

The proposed technique to code the I-contours (i.e. the reference contours i_0 and i_1) is summarized in Fig. 2.7. The computation of the probability distribution for each symbol is done using a window of previously encoded symbols of length N_0, using Eq. 2.7 as the estimator for the most probable direction, and the adaptive statistical model described in 2.2.2. The decoder for the I-contours is exactly the same as the encoder: using a window of already received symbols, the probability distribution for the next symbol can be computed and the bit-stream correctly decoded.

The lossless contour coding technique for B-contours is summarized in Fig. 2.8: to code the curve b we first use the reference curves i_0 and i_1 to make an elastic prediction. Then the dynamic time warping is performed to establish a correspondence function between b and the selected elastic deformation $\widetilde{b}_s$. The curve $\widetilde{b}_s$, along with the correspondence function, provides a useful context to estimate with more accuracy the probability distribution for each symbol of b. The symbols of the curve to code b and their probability distributions are then fed to an arithmetic encoder that produces the bit stream BS_b.

The decoder side of the contour coding technique for B-contours is summarized in Fig. 2.9. The first step is again the elastic prediction from the reference contours i_0 and i_1, which is used in conjunction with the correspondence function and the parameters N_0, N_f, ρ and s, to estimate for each symbol the same probability distribution as the encoder. The bit stream BS_b is thus decoded into the symbols that compose the curve b.

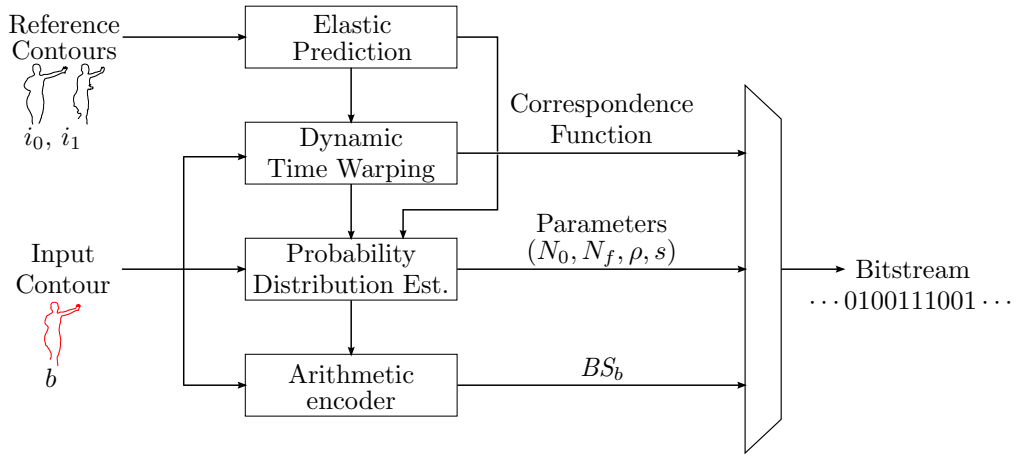


Figure 2.8: Synthetic scheme of the coder for the lossless contour coding technique for B-contours.

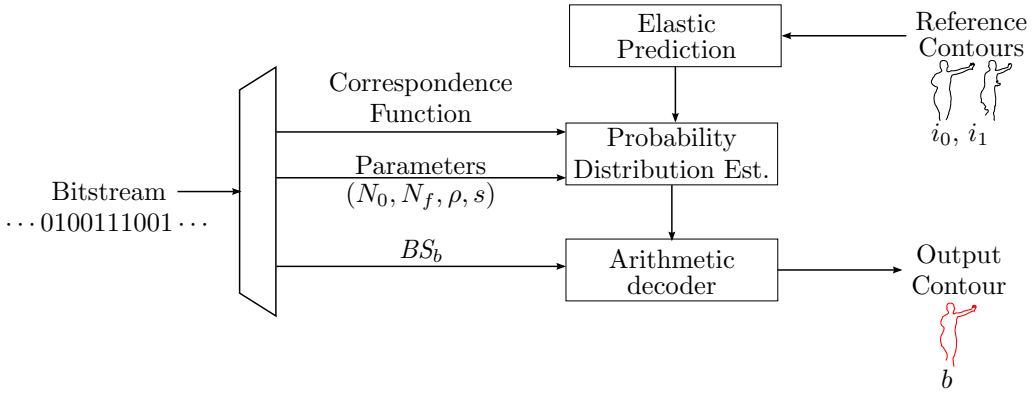


Figure 2.9: Synthetic scheme of the decoder for the lossless contour coding technique for B-contours.

2.3 Experimental results

The proposed method has been evaluated using the multiview sequences: *ballet*, a real world sequence provided by Microsoft Research with resolution 1024×768; *mobile*, a synthetic sequence provided by Philips), its resolution is 720×540; *lovebird*, a real world sequence provided by the ETRI/MPEG Korea Forum, whit resolution 1024×768; and *beergarden*, a sequence where the two main actor are real and the background is synthetic, whose resolution is 1920×1088 and it was provided by Philips. We encoded the curves corresponding to the main object in the depth sequences for a fixed time instant or view. To test our algorithm we used also masks extracted from the real world monoview sequences such as *stefan* and *foreman*, which have CIF and QCIF resolutions: 352×288 and 176×144, respectively.

The curves for *ballet*, *mobile*, *lovebird* and *beergarden* depths were obtained using the Canny edge detector [Can86]. Depth maps are not as complex as texture images and the use of the Canny edge detector produced very good results in our test cases. We extracted the contours of very precise segmentation maps for the sequences *stefan* [CBS09], [CB09] and *foreman* [Lin].

The coding scheme concentrates almost all the computational load at the encoder side. In particular for the B-contours the choice of the values of (N_p, N_f, ρ, s) can be made by trying out every possible combination and selecting the one that gives the least bit-rate. This full search method, however optimal, is extremely costly in terms of time and complexity. We thus introduced a greedy algorithm that optimizes one variable at a time to find a sub-optimal solution.

2.3.1 Coding of I-contours and B-contours

To code the I-contours we replaced the linear regression (LR) with the more effective average direction (AD) in the technique described in [DCF12], whereas for the coding of the B-contours we can take advantage of the context provided by the elastic curves (EC), thus achieving better coding gains. In Tab. 2.1 are shown the results for a set of images taken from the test sequences. The reported values take into account the side information cost needed by each method. We notice that just altering the direction extraction method from LR to AD the average coding cost is reduced by 4.88%, while passing from AD to AD with EC context leads to a reduction of 1.82%, for a total reduction of 6.53%. If on the other hand we use the LR with the EC context, the rate reduction is 2.25%.

2.3.2 Side information cost

We will now account for the cost to transmit the four parameters $s^\star$, $N_p^\star$, $N_f^\star$ and $\rho^\star$, as well as the correspondence function.

	linear regression	average direction	linear regression + EC context	average direction + EC context
ballet	1428.20	1375.20	1386.94	1338.78
beergarden	1687.67	1610.33	1656.67	1575.58
lovebird	1424.82	1370.73	1382.08	1341.73
mobile	711.71	642.29	697.15	639.12
foreman	411.73	396.82	398.15	390.20
stefan	745.33	701.00	738.33	700.00
average	*1068.24*	*1016.06*	*1043.22*	*997.57*

Table 2.1: Coding results (in bits) for the different contributions of the developed tools to the technique proposed in [DCF12], applied to object contours. Two different methods to extract the probable direction from a set of points: linear regression, and average direction, without and with EC context.

The correspondence function is the result of a linear approximation and experiments show that 10 bits are enough to code the two parameters of the straight line with sufficient precision.

For $N_p^\star$ and $N_f^\star$ we decided to use 1 bit for the range of N_p ($\{5, 6\}$), and 2 bits for the range of N_f ($\{6, 7, 9, 11\}$), once again based on experimental results. On the other hand the optimal ρ has a wider range of values, and we observed that typical values can be represented by the set $\{6.6 + k\Delta\}$, with $\Delta = 0.1$ and $k = 0, \ldots, 31$, for a cost of 5 bits.

We observe that the accuracy of the representation of the position on the geodesic $s^\star$ has a quite large influence on the coding efficiency. Using for example 2 bits to code the possible positions we have 4 curves to use as side information to code the curve b, using 3 bits leads to 8 curves, in general using a fixed length representation with b_s bits permits us to choose among 2^{b_s} different curves. The chance of finding a good matching curve to use as a side information increases with b_s, but for every bit added the complexity doubles, and we are increasing the cost of the representation too. If some values are more probable than others it is worth to consider a variable length code to reduce the cost of the representation.

We thus used an experimental approach and compared two different ways to code $s^\star$: a fixed length coding, with length from 2 to 10, and an Exponential-Golomb code. In Tab. 2.2 we see that, performing the average on different curves and on different sequences, a fixed-length coding leads to better performance if the number of bits used for the representation of $s^\star$ approaches to 6 or more. To compare the proposed method to other techniques we choose the best performing 10 bits fixed length coding.

For the I-contours we only have to transmit as side information the parameters $N_p^\star$ and $\rho^\star$, corresponding to an overall cost of 6 bits. On the other hand the decoding of the B-contours needs the correspondence function as well as the four parameters $N_p^\star$, $N_f^\star$, $\rho^\star$ and $s^\star$, corresponding to an overall cost of 28 bits.

Sequence	2 bits	3 bits	4 bits	5 bits	6 bits	···
ballet	1338.20	1329.00	1323.80	1323.80	1323.40	
beergarden	1573.33	1565.33	1559.33	1560.33	1559.00	
lovebird	1370.45	1347.18	1336.73	1334.18	1331.27	
mobile	638.29	635.29	635.43	635.57	634.57	
foreman	390.27	388.27	388.45	388.91	387.00	
stefan	705.00	695.67	693.00	693.00	691.67	
average	1002.59	993.46	989.46	989.30	987.82	

Sequence ···	7 bits	8 bits	9 bits	10 bits	Exp-Golomb
ballet	1321.40	1319.80	1318.20	1316.20	1320.80
beergarden	1559.67	1554.67	1554.33	1554.00	1559.00
lovebird	1328.73	1328.18	1324.73	1323.91	1332.09
mobile	634.86	635.71	635.00	634.71	635.71
foreman	386.82	387.09	386.55	387.09	388.09
stefan	689.33	688.33	689.33	690.00	694.00
average	986.80	985.63	984.69	984.32	988.28

Table 2.2: Average coding cost (in bits) for different ways of coding $s^\star$: fixed length coding up to 10 bits and Exp-Golomb.

2.3.3 Greedy algorithm

Resting on experimental results for each variable we selected a range of typical values. We initialize the greedy algorithm (GA) with a starting point $(N_{p0}, N_{f0}, \rho_0, s_0)$, corresponding to the solution in which every value is the closest to the center of the coded range, and then the GA optimizes the variables in the order N_p, N_f, ρ, s. Keeping fixed N_{f0}, ρ_0 and s_0 the GA runs the proposed technique to select the N_p that minimizes the bit rate. Once $N_p^\star$ has been found the algorithm starts again to optimize N_f from the point $(N_p^\star, N_{f0}, \rho_0, s_0)$. Then again for ρ and s, until it reaches the solution $(N_p^\star, N_f^\star, \rho^\star, s^\star)$.

The search order for the parameters is set according to our observations of the optimum parameter distributions after the full search for many sequences: $N_p^\star$ has a very peaked distribution, so the selected value after the first step of the GA should actually be the best one. $N_f^\star$, on the contrary, has an almost uniform distribution, thus making difficult to locate the best value; it is however required to select it before $\rho^\star$, because the best value of ρ is influenced by the selected values of N_p and N_f. Still based on our observations, the last parameter to select is s.

As we can see in Tab. 2.3, the average loss with respect to the full search is 0.82%, but the number of calculations the encoder has to do is approximately reduced by a factor of 250.

Regarding the overall time needed for the coding of a curve, we can distinguish fixed and variable time contributions. The elastic estimation is fixed but we can decide how many frames to leave in between the two reference ones. On the other hand the execution time for the choice of the parameters can vary greatly: a full search is very costly, and even if the greedy algorithm speeds up the whole process, one can also decide to keep the same

Sequence	Full Search	Greedy Algorithm
ballet	1309.60	1316.20
beergarden	1548.00	1554.00
lovebird	1305.18	1323.91
mobile	631.43	634.71
foreman	380.82	387.09
stefan	684.00	690.00
average	976.51	984.32

Table 2.3: Average coding cost (in bits) for the full search and the greedy algorithm.

parameters (or a subset of the parameters) for a certain number of frames.

2.3.4 Comparisons

We compare our technique to various methods to code the differential chain code of the contours: Adaptive Arithmetic Coder (AAC), Context Based Arithmetic Coder (CBAC), and the technique proposed in [DCF12].

In the compression of B-contours we achieve gains up to 10% compared to the method of [DCF12], but to make a fair comparison we have to consider for our technique both I-contours and B-contours of the GOP structure. To study the influence of the GOP structure on the coding performance we used the following:

- IBIBIB... is very effective for the B-contours, since the two reference frames are very close and the prediction is very accurate, but the I-contours have a non negligible cost on the final outcome. Using this GOP structure produced in our experiments an average bit rate of 1000,19 bits per contour;
- IBBIBB... and IBBBIBBB... are non-hierarchical structures with fairly distant I-contours, they produced an average bit rate of 1004.03 and 1010.28 bits per contour respectively;
- $I_1B_1B_2B_3I_2$... is a hierarchical structure with 5 frames in the GOP, in which B_2 is predicted using I_1 and I_2, B_1 using I_1 and B_2. It produced an average bit rate of 997,57 bits per contour.

The hierarchical structure proved to be the most effective GOP structure: the cost of the I-contours is low and the elastic prediction for B_2 is just slightly less accurate than the ones for B_1 and B_3. This result is expected, given the previous study on depth map compression with traditional hybrid techniques that shows the importance of the prediction order [MPP08].

In Tab. 2.4 the average results for the test sequences are shown, and in every case our technique performs better than the references. The overall average gain with respect to the second best coding technique in the group, the one proposed in [DCF12], is 6.53%. If

Sequence	# symbols	JBIG2	AAC	CBAC	[DCF12]	Proposed	Gain
ballet	1125.00	3968.00	1715.40	1585.60	1428.20	**1338.78**	6.26%
beergarden	1369.00	4226.67	2052.33	1882.67	1687.67	**1575.58**	6.64%
lovebird	1302.09	3153.45	1886.55	1740.36	1418.82	**1341.73**	5.43%
mobile	661.86	1915.43	837.14	696.00	711.71	**639.12**	10.20%
foreman	330.00	1776.73	517.18	522.91	411.73	**390.20**	5.23%
stefan	565.00	2106.67	859.33	877.33	745.33	**700.00**	6.08%
average	*892.16*	*2857.82*	*1311.32*	*1217.48*	*1067.24*	*997.57*	*6.53%*

Table 2.4: Average coding cost (in bits) for various sequences in the view domain (*ballet*) and in the time domain (*mobile*, *lovebird*, *beergarden*, *stefan*). The tested methods are: JBIG2, Adaptive Arithmetic Coder (AAC), Context Based Arithmetic Coder (CBAC) with 1 symbol context, the one proposed in [DCF12], and the proposed technique (all the side information cost accounted). In the last column are reported the gains of the proposed technique over the other best performing one in the group.

we consider instead the gain of the proposed technique with respect to JBIG2, which is not optimized for this kind of data but has been chosen to code the boundary information in [GLG12], it is 65.09%. Over other standard techniques, such as AAC and CBAC (with one symbol context, the best choice in our tests) the average gains are of 23.93% and 18.06%, respectively.

2.4 Conclusions

In this chapter, we have described a novel technique for lossless coding of object contours for MVD, originally proposed in [CCPP14]. Using elastic deformation between two reference contour curves, we obtained a useful side information for the coding of the actual contour. The price paid with the coding cost for the side information is fully rewarded with significant gains with respect to the reference techniques and to the state of the art. We have also improved the technique described in [DCF12] by substituting its prediction method with the average direction method.

The technique and results presented in this chapter are published in the journal article [CCPP14]. So far only a monodimensional elastic interpolation has been considered, but we expect a more precise estimation if we can take into account 4 or 8 reference curves from different views and different times, thus leading to further improvements of the technique. This will be the subject of further study.

In the next chapter we will examine some application of this contour coding technique to depth map coding, showing that this approach can give interesting results, even if compared to state-of-the-art techniques, such as HEVC Intra.

Chapter 3

Shape-based depth map codecs

Contents

Depth maps can be represented by means of grayscale images and the corresponding temporal sequence can be thought as a standard grayscale video sequence and thus compressed with standard video coding techniques. However depth maps have different properties from natural images: representing the distance from the camera and the objects in the scene, they present large areas of smooth surfaces (high level of redundancy) separated by sharp edges. In terms of compression performance, arguably better results can be obtained by exploiting ad-hoc approaches tailored to the specific properties of this representation. In this chapter we present two novel coding schemes where the depth data is represented by a set of contours defining the various regions together with a compact representation of the values inside each region.

As we will see in Section 3.1, segmentation information has been widely used for depth compression but the coding of the segments shape is critical and most available approaches are not able to propagate the information across multiple frames. In order to exploit

the temporal or inter-view redundancy of object contours and code more efficiently the contour information, we use for the two presented coding schemes the lossless contour coding technique with elastic deformation of curves described in Chapter 2. After the main discontinuities have been captured by the contour description, the depth field inside each region is rather smooth.

We proposed and tested two different techniques for the coding of the depth field inside each region. The first one is very simple and uses the Shape Adaptive Wavelet Transform, followed by SA-SPIHT (Set Partitioning In Hierarchical Trees). The second one on the other hand is more complex and uses a 3D surface prediction algorithm in order to obtain an accurate estimation of the depth field from the contours and a subsampled version of the data. Follows an ad-hoc coding strategy for the low resolution data and the prediction residuals. Experimental results prove how the proposed approaches are able to compete with, and in some cases outperform, the state of the art. Moreover it is important to notice that in the proposed techniques is performed a lossless contour coding: preserving the sharp edges between the various surfaces is a relevant property since the positive impact of contour-preserving compression techniques on synthesized images, obtained through DIBR (Depth Image Based Rendering) approaches [Feh04], has been confirmed by means of subjective tests. A detailed discussion on the subject is reported in Chapter 4.

The outline of the chapter is as follows. We first make a short overview of different codecs specifically targeted at depth maps in Section 3.1, then we will describe the proposed techniques EC+SA-SPIHT and EC+3D surface prediction, along with experimental results, in Sections 3.2 and 3.3, respectively.

3.1 Related work

The vast majority of the techniques specifically targeted at depth map compression exploits the key idea that depth maps are made of smooth regions divided by sharp edges. To attempt a simple classification of the many approaches proposed in the literature we can divide them into two main categories: *block-based* coding techniques and *non block-based* coding techniques [CPD13].

Block-based coding techniques

Among the block based coding techniques we can take as example the work by Morvan *et al.* [MFdW07]: a quad-tree decomposition divides the depth image into blocks of different size and each block is approximated with a function. To model the signal in each block are used a constant function, a linear function, a piece-wise constant function (wedgelet), or a piece-wise linear function (platelet), if the approximation provided by these function for the block is not good enough, the block is then divided into four sub-blocks, and the modelling functions are checked again against the sub-block. The process is repeated until

each leaf of the quad-tree is approximated. This technique has been further explored in [MMS+08] and enanched and refined in [SBB11].

To better adapt to the content, in particular to the sharp edges that characterize depth maps, many transforms have been proposed. In [SKN+10] Shen *et al.* describe edge-adaptive transforms (EAT) that can be used for any edge structure. For each block the first step is to locate the edges in between adjacent pixels. Then the pixels of the block are mapped on a graph, where each pixel is connected to its neighbours unless they are separated by an edge. Finally an EAT can be applied on this graph. The main idea is to have sets of pixels separated by an edge in different graphs, thus avoiding filtering across the edges by applying a transform only to the values of the connected pixels.
The idea of graph-based transform (GBT), along with transform domain sparsification (TDS) has been discussed in [CKO+11] by Cheung *et al.* Again, to code blocks that contain relevant edges in depth maps, the GBT is used: avoiding the filtering across the edges, that results in large non-zero high frequency coefficients, compensate for the graph coding overhead. TDS was proposed to define per-pixel sensitivity of the synthesized view to errors in the depth map. After computing this metric a search is performed for a sparse depth signal in an orthogonal transform domain, achieving compression gain introducing a controlled distortion in the synthesized view. The combination of GBT and TDS is used for the blocks of the depth image, under a unified optimization framework.

The coding of depth videos has also been considered, e.g., in [DCF12] the idea that pixels with similar depth have similar motion is exploited: whether a block presents an edge, it is divided in two sub-blocks before performing motion estimation, and the estimation is done separately for each sub-block. In this case the overhead caused by the need of lossless coding of the contour is compensated by the very small prediction residuals. Another possibility is to exploit adaptive interpolation schemes able to correctly handle edge regions, for example the approach in [OYVH09] by Oh *et al.* subsamples the depth information and then reconstructs it with an adaptive interpolating filter.

Non block-based coding techniques

Traditional block-based coding paradigms, widely used for natural images, are less effective for depth map coding as a consequence of their distinctive characteristics. As a result many ad-hoc techniques that do not follow a block-based coding scheme have been proposed.

An example for a novel representation of depth maps was proposed by Farin *et al.* in [FPdW07]. To obtain a high compression for an elevated number of views triangular meshes are produced to provide a 3D scene representation. To avoid evident artefacts along the borders in view synthesis, the mesh is constructed to have no triangle placed along the discontinuities, and to each node two values are assigned to better model depth discontinuities.
The same approach of just one 3D representation for a large number of views was adopted

by Maceira *et al.* in [MMRH15], only instead of a mesh, a 3D plane-based representation is used. The similarity between the texture and the depth signals is used in order to build a partition of the depth, then a plane fitting is performed to approximate each region.

The aforementioned techniques, although well adapted to the sharp edges of depth images, are not able to provide a pixel-wise accuracy of the object boundaries, fundamental for a high quality synthesis of novel views. Another possible solution is thus to exploit a segmentation of the depth map in order to decompose it into a set of homogeneous regions that can be represented by simple models or low resolution approximations. Examples of approaches belonging to this family are [SSO09], [ZC09],[MC10], [Jag11], and [GLG12]. The segmentation or edge information they all use can have a relevant impact on the total bit-rate, for this task efficient contour coding schemes are needed. Typical contour coding techniques rely for contours on chain coding and differential chain coding [KKM+98], on polygon approximation [KBE00], on Hausdorff-distance constrained coding [HMS13a], and more frequently on JBIG [ISO93] to encode the whole segmentation map. Even if they are quite effective, these techniques do not exploit the temporal redundancy of contours of the same objects in different time instants or views.

Sanchez *et al.* in [SSO09] propose a wavelet-based scheme, in combination with an edge-preserving lifting scheme to avoid filtering across the edges. The cost of sending the contour map to the decoder is balanced with the reduction of larger coefficients generated typically near the edges. The solution proposed by Zanuttigh and Cortelazzo in [ZC09] still exploits segmentation, but followed by a linear prediction scheme: a subsampled version of the depth map is used to perform a 3D surface prediction for the points inside the regions recorded on the segmentation map. A further refinement layer encodes the residuals with a JPEG2000 coder. Milani and Calvagno in [MC10] proposed a scheme that exploits a scalable decomposition into progressive silhouettes: the regions obtained by merging of the segmentation elements are approximated with their relative average, this representation provides a first approximation of the depth map. Again a refinement layer codes the residuals with a standard H.264/AVC coder. In a similar fashion, in [Jag11], Jager uses piecewise-linear functions to approximate the surfaces, which are not necessarily parallel to the image plane but are generally characterized by linear gradient in depth images. Another method, proposed by Gautier *et al.* in [GLG12], transmits to the decoder the map of the contours, the values of the pixels along the edges, and a subsampled version of the depth map. The decoder uses the received depth values as seeds to perform an interpolation by means of a diffusion algorithm.

The idea of exploiting crack-edges is used in [TSA14] by Tabus *et al.* Crack-edges are defined as the lines that separate pixels of different values and are used to define regions of constant depth value. To have a efficient representation of the depth map with these elements redundancy between adjacent regions is exploited, as well as edges positions. Crack-edges are used also in the approach proposed by Mathew *et al.* in [MTZ13], where a breakpoint field representing crack-edges is exploited, in order to avoid filtering across the

edges in the wavelet reconstruction.

Other techniques

Depth maps and texture images are highly correlated, so it is possible to use the texture component to achieve high compression gains for the coding of depth maps. To give some examples, color data has been used to assist the compression of depth information in the works [FLWM15], [MZZF11], [MC11a], [HMS13b] and [GBG15]. In addition, the approach in [MC11b] performs an object classification on the scene and adapts the depth coding strategy according to the motion characteristics.

The approach of [ZJZ+09], while coding jointly texture and depth, also adopts a segmentation of the depth image followed by region-based coding scheme because of the impact that depth values along object borders have on the rendered virtual view images.

3.2 Depth map coding with elastic deformation of contours and SA-SPIHT

The depth map coding technique presented in this section has been proposed in our paper [CCPP14], and even though its main goal was to show the potential of the lossless contour coding based on elastic deformation, we found that a relatively simple codec based on this tool may be competitive with the state of the art. This may be considered a validation of the elastic deformation-based approach.

3.2.1 Technique description

We resort to an existing object-based technique since it can immediately benefit from an improved contour coding method. Despite the simplicity of the approach the results are satisfying, due to the nature of the data we want to compress. In summary, the new object-based compression technique is composed of:

- lossless coding of the segmentation map. We apply the the lossless contour coding technique described in Chapter 2 with the contours of the objects. We use a hierarchical GOP structure $I_1B_1B_2B_3I_2$, in which B_2 is predicted using I_1 and I_2, B_1 using I_1 and B_2, and so on, as described in 2.3.4. This part provides a lossless coding with inter-frame prediction;

- Shape Adaptive Wavelet Transform (SA-WT) for each object, followed by SA-SPIHT (Set Partitioning In Hierarchical Trees), followed by an arithmetic coder [CPVZ04]. We remark that for the inner part of the objects we thus have an entirely "Intra" technique. We have chosen this technique because it is reasonable and simple, and it complements perfectly with our lossless coding technique.

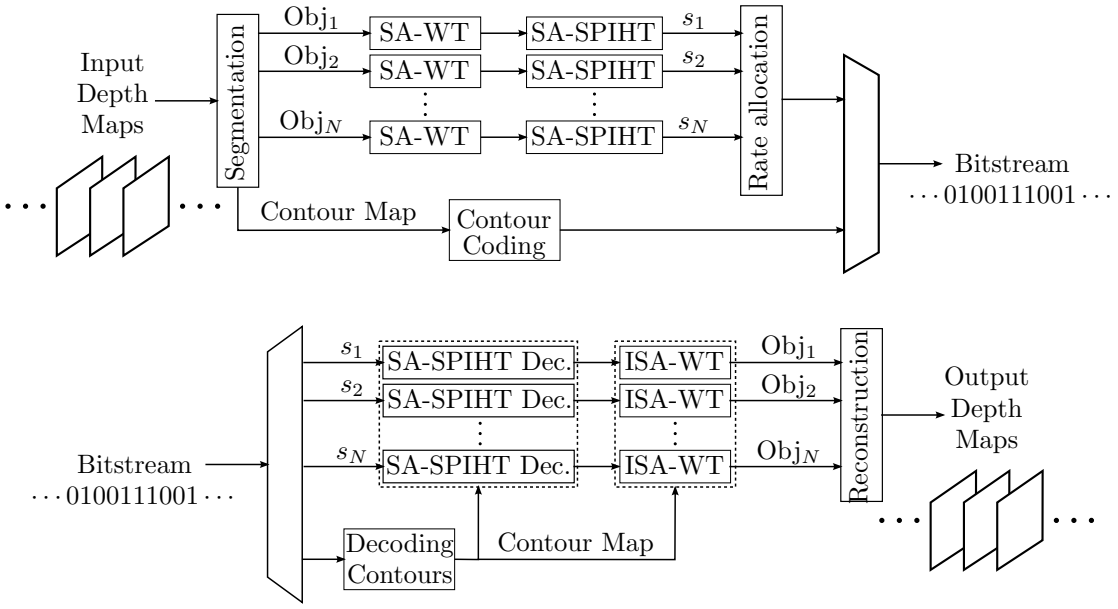


Figure 3.1: Coder and decoder schemes of the proposed technique EC + SA-SPIHT.

A synthetic scheme of the proposed depth map coding technique is shown in Fig. 3.1. While the lossless contour coding technique has been widely discussed in Chapter 2, a brief description of the technique to code the inner parts of the object is provided here.

Once the depth map has been segmented in homogeneous regions, we use a transform coding technique based on the shape-adaptive wavelet transform: SA-WT, using the algorithm of Li and Li [LL00], whose good performance and simple implementation make it adopted in the MPEG-4 standard. The number of level decomposition is set to 5 and Daubechies 9/7 biorthogonal filters are used. The most relevant features of the SA-WT are that no new redundancy is introduced and the number of coefficients is the same as the number of pixels in the original object. Moreover the spacial relationship among pixels is retained and there are no new frequencies introduced in the transform domain. Finally, for rectangular objects the SA-WT is equivalent to the regular WT.

The SPIHT [SP96] algorithm follows the SA-WT, in the shape-adaptive extension proposed in [CPVZ04]. The SPIHT algorithm is a bit-plane coder of the wavelet coefficients that exploits the similarities across the sub-bands in a wavelet decomposition of an image. For each bit-plane there are essentially two tasks, locating the significant bits, and specifying their value and sign. The most important coefficients are coded first, and increasingly all the coefficient to obtain a more refined copy of the original image can be coded. The shape-adaptive version of SPIHT proposed in [CPVZ04] is very similar to the basic algorithm with the differences that only active nodes, that is nodes belonging to the support of the SA-WT transform, are considered, and that the tree of coefficients has a single ancestor in the lowest frequency band.

After the encoding of the coefficients of the different objects, the RD curves of all objects are analysed to allocate among them the available bits for a desired encoding rate. This step is necessary for a object-based coder, and although RD criteria are usually used for bit allocation, there is another degree of freedom as the allocation could be done also according to different criteria.

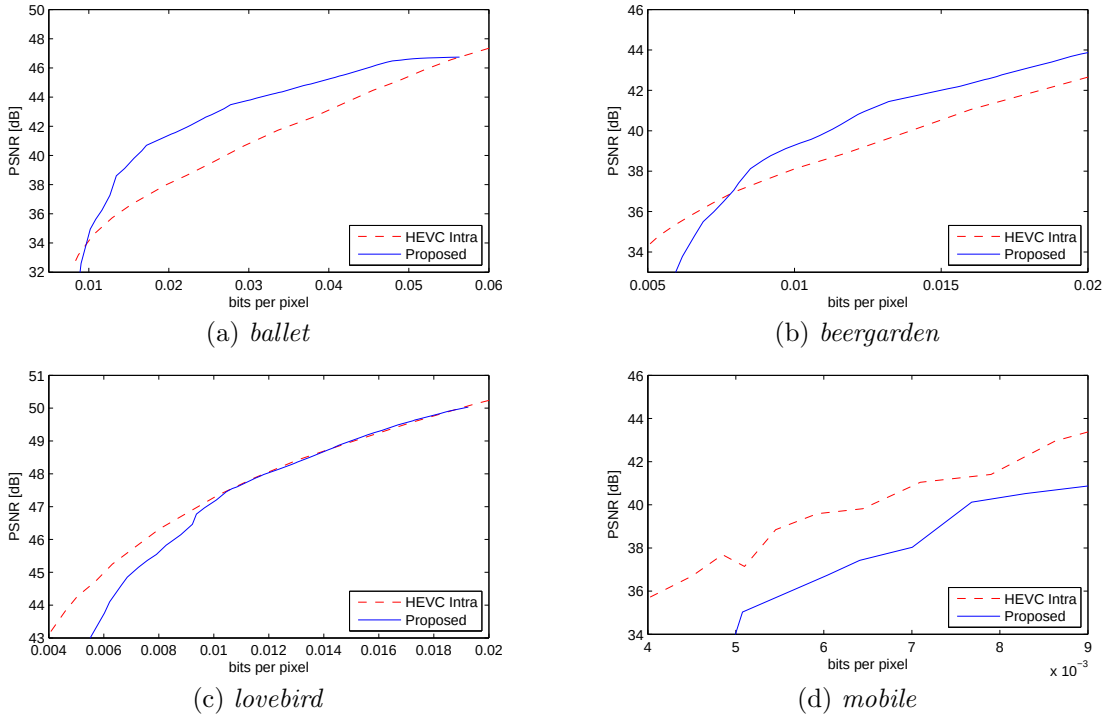
3.2.2 Experimental setting

To make a meaningful comparison we used HEVC Intra to compress the depth maps. We believe that the comparison is fair because in the case of HEVC Intra the arithmetic coder for the lossless coding part take advantage of the context updating, and there is no temporal prediction. Likewise, in our technique, the lossless coding part exploits the temporal redundancy, while the object coding is totally "Intra". Moreover it would not be fair to make a comparison with HEVC Inter because we have no temporal prediction for the objects, neither would be easy to develop an object-based coder with temporal prediction.

Once defined the compression technique, we use the decoded depths to synthesize new views and make a comparison with the images generated by the uncompressed depth maps. Given any two adjacent views of the multiview sequence, we generated three equally spaced synthetic intermediate views. To test our depth maps compression technique we used the multiview sequences *ballet*, *beergarden*, *lovebird* and *mobile*, synthesizing 6, 15, 30 and 54 frames, respectively. In order to assess the quality of the virtual views, we compared them to synthesized images obtained by applying the same DIBR algorithm to the uncompressed depths and views, and thus we obtained the RD points related to our techniques and to HEVC Intra.

3.2.3 Results

A first comparison between the proposed technique EC + SA-SPIHT and HEVC Intra can be done by computing the PSNR of the compressed depth maps. For the sequence *ballet* there is a significant gain of the proposed technique over HEVC Intra, up to 3.5 dB, in the range of 0.01 to 0.05 bpp, as reported in Figure 3.2(a). For the sequence *beergarden*, starting from 0.075 bpp, there is an almost constant gain of 1.3 dB of the proposed technique over HEVC Intra, as reported in Figure 3.2(b). The two techniques perform equally good for the sequence *lovebird*, where the proposed technique reaches the same PSNR of HEVC Intra starting from 0.01 bpp. The Figure 3.2(c) depicts the behaviour of the two codecs in the bit-rate range. Finally, there is a significant advantage of HEVC Intra over the proposed technique for the sequence *mobile*, as shown in Figure 3.2(d). This is due to the fact that the edges are already blurred in the original depth map, reducing the effectiveness of coding strategies based on the assumption of sharp edges between different regions. These comparisons are informative about the effectiveness of the proposed coding strategy,



(a) *ballet* (b) *beergarden*

(c) *lovebird* (d) *mobile*

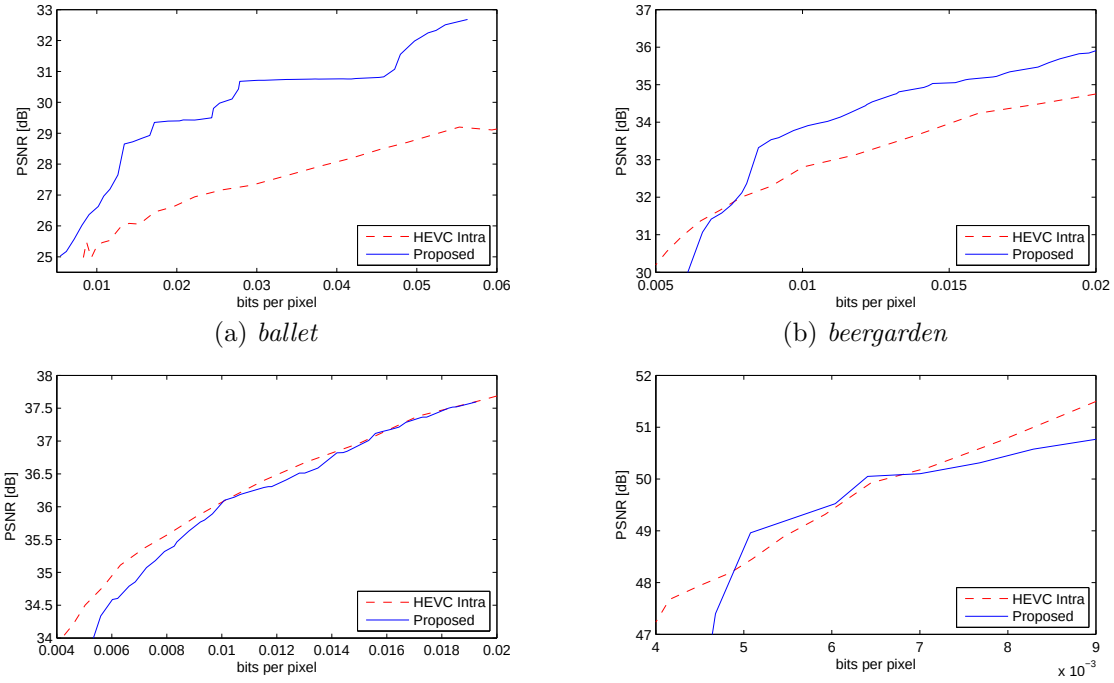
Figure 3.2: Depth map coding technique EC + SA-SPIHT: PSNR of the compressed depth maps for the sequences *ballet* (a), *beergarden* (b), *lovebird* (c), and *mobile* (d).

but they do not provide any clue about the validity of the approach for DIBR use.

The compressed depth maps are used in conjunction with the DIBR engine to synthesize novel views. The synthesized images obtained with depth maps compressed with the proposed technique are compared to the synthesized images obtained with depth maps compressed with HEVC Intra. In order to take into account the artifacts introduced by the DIBR software, the reference for this kind of images are the synthesized images obtained with uncompressed depth maps [EYUHG10].

PSNR and SSIM are computed and showed in Figure 3.3 and Figure 3.4, respectively. In particular, concerning the PSNR of the synthesized images, the Bjontegaard metric [Bjo01], computed on the RD points showed in the graphs, points out that the proposed technique outperforms HEVC Intra for *ballet*, *beergarden* and *mobile*, achieving respectively +2.6 dB, +0.16 dB and +0.88 dB, while the PSNR was practically identical for *lovebird*.

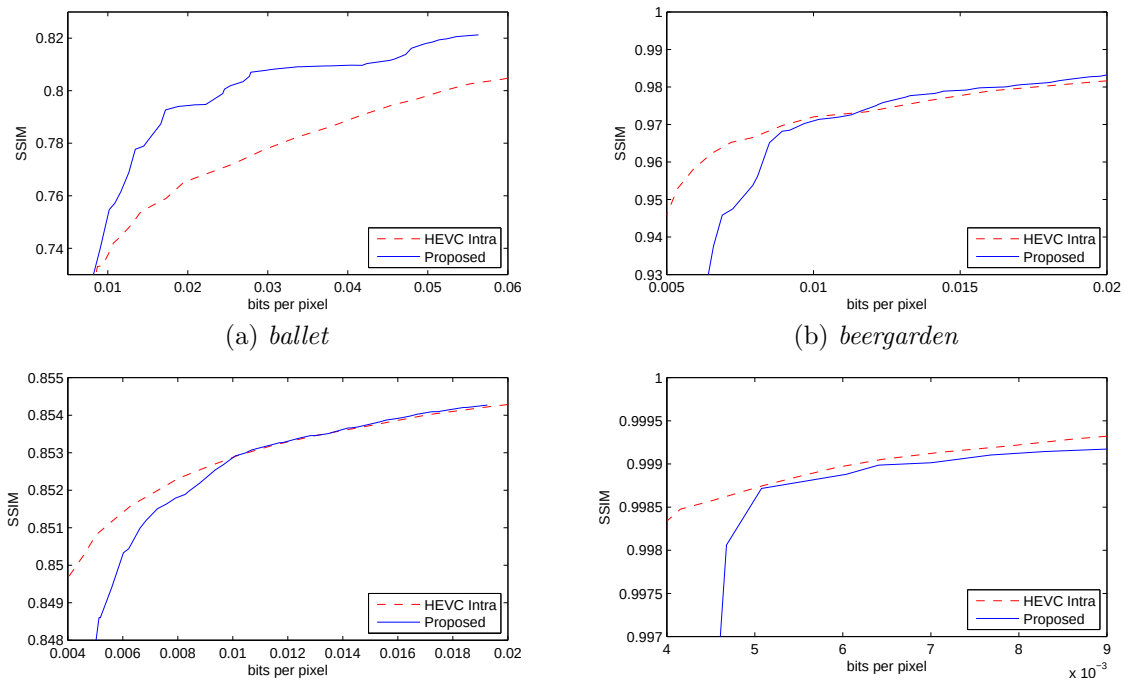
The SSIM metric, more refined than the PSNR as it takes into account some perceptual phenomena of the human visual system, shows that for the sequences *beergarden*, *lovebird* and *mobile*, the performance of the two codecs is very similar, as shown in Figure 3.4(b), (c) and (d). The two codecs however introduce very different artifacts. The HEVC Intra codec introduces typical fragmentation artifacts along the borders in the synthesized view, as shown in Figure 3.5(b1) and put in evidence in the detail in Figure 3.5(b2). On the other hand, the artifacts produced by the proposed EC + SA-SPIHT codec are more subtle



(a) *ballet* (b) *beergarden*

(c) *lovebird* (d) *mobile*

Figure 3.3: Depth map coding technique EC + SA-SPIHT: PSNR of the synthesized images obtained using the compressed depth maps for the sequences *ballet* (a), *beergarden* (b), *lovebird* (c), and *mobile* (d).



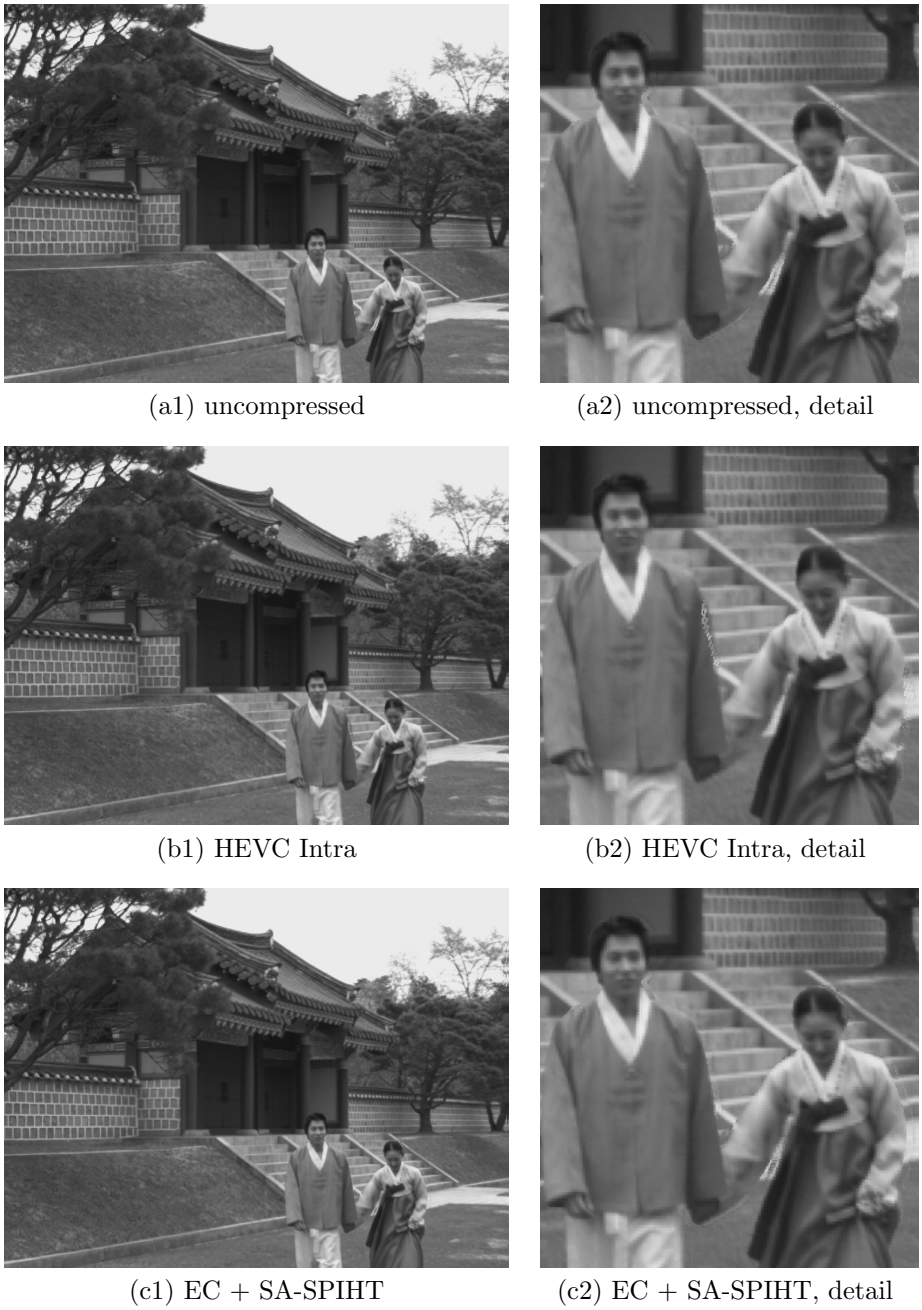
(a) *ballet* (b) *beergarden*

(c) *lovebird* (d) *mobile*

Figure 3.4: Depth map coding technique EC + SA-SPIHT: SSIM of the synthesized images obtained using the compressed depth maps for the sequences *ballet* (a), *beergarden* (b), *lovebird* (c), and *mobile* (d).

to detect. Due to the coarse representation of the geometric information of the scene, the object textures or the background may be slightly misplaced, as shown in Figures 3.5(c1) and (c2). This kind of errors affect heavily the PSNR, while a human observer may not perceive any difference if the synthesized images obtained using depth maps compressed with the proposed method are compared to the reference, synthesized images obtained using uncompressed depth maps. The reference image for the example of Figure 3.5 are reported as Figure 3.5(a1) and (a2).

These results, however pertinent to quite specific test conditions (limited set of sequences, limited range), show that our technique can perform at least as well as HEVC Intra, or even better depending on the sequence: these results imply that the proposed approach is worth considering. This is even more relevant in sight of the fact that in general, object-based coding techniques achieve not very good results in image compression. It has been shown that the cost of lossless contour coding is one of the elements that undermine these techniques the most [CPPV07].



(a1) uncompressed (a2) uncompressed, detail

(b1) HEVC Intra (b2) HEVC Intra, detail

(c1) EC + SA-SPIHT (c2) EC + SA-SPIHT, detail

Figure 3.5: Sequence *lovebird*: different artifacts introduced by HEVC Intra (b) and the proposed technique EC + SA-SPIHT (c). The reference image (a), obtained using uncompressed depth maps, is reported to show the artifacts introduced by the rendering. HEVC Intra is used at QP 44, and the bit-rate of the technique EC + SA-SPIHT is set to match exactly the one of HEVC.

3.3 Depth map coding with elastic deformation of contours and 3D surface prediction

The technique presented in Section 3.2.1 is very effective when the objects have a distance from the camera that ranges from medium to high. In that case the coarse representation of the geometry of the inner part of the objects do not cause any problem in the synthesis of novel views. On the other hand, when an object is very close to the camera, it can have a uneven depth region, which would require a large amount of bits to be coded with an acceptable quality for synthesis purposes. We therefore developed a novel coding scheme where the depth data is represented by a set of contours defining the various regions together with a compact representation of the values inside each region. A 3D surface prediction algorithm is then exploited in order to obtain an accurate estimation of the depth field from the contours and a subsampled version of the data. The low resolution data and the prediction residuals are compressed with ad-hoc strategies.

3.3.1 Technique description

The general workflow of the proposed approach is shown in Figure 3.6. The first step is the segmentation of the depth maps in order to extract the main objects and structures in the scene. The segment contours are then compressed using the approach described in [CCPP14] and the average depth value in each segment is stored. The depth maps and the segmentation data are subsampled and the difference between the subsampled representation and the segment averages is compressed using a differential prediction scheme followed by the HEVC coder. Then a surface prediction algorithm derived from [ZC09] is used to predict the input depth maps from the subsampled data and the contours. Finally, the residuals between the prediction and the input depth map are lossy compressed using the HEVC coder.

After the segmentation and the coding of the segment contours, the next step is the subsampling of depth data. The depth map is simply subsampled according to a regular grid (see Fig. 3.7c) and the average depth values in the segments are subtracted from the subsampled data. The sampling factor Δ is selected according to the resolution and the amount of detail of the depth data. In the experimental set-up presented in 3.3.2, sampling grids of 16×16 or 32×32 were used. The subsampled data is basically a low resolution depth map of size (W/Δ) by (H/Δ), where W and H are the dimensions of the original depth map. This information is compressed in lossless way in order to prevent blurring on the edges and averaging between neighboring samples of different segments, which would make useless the benefits of the segmentation step. In order to obtain a high coding gain, the samples of the first depth image (Intra depth frame) are first scanned on a raster order and converted into a sequence of couples (r, l) according to a run-length coding strategy. Then, both runs and lengths are coded using a Lempel-Ziv 78 algorithm,

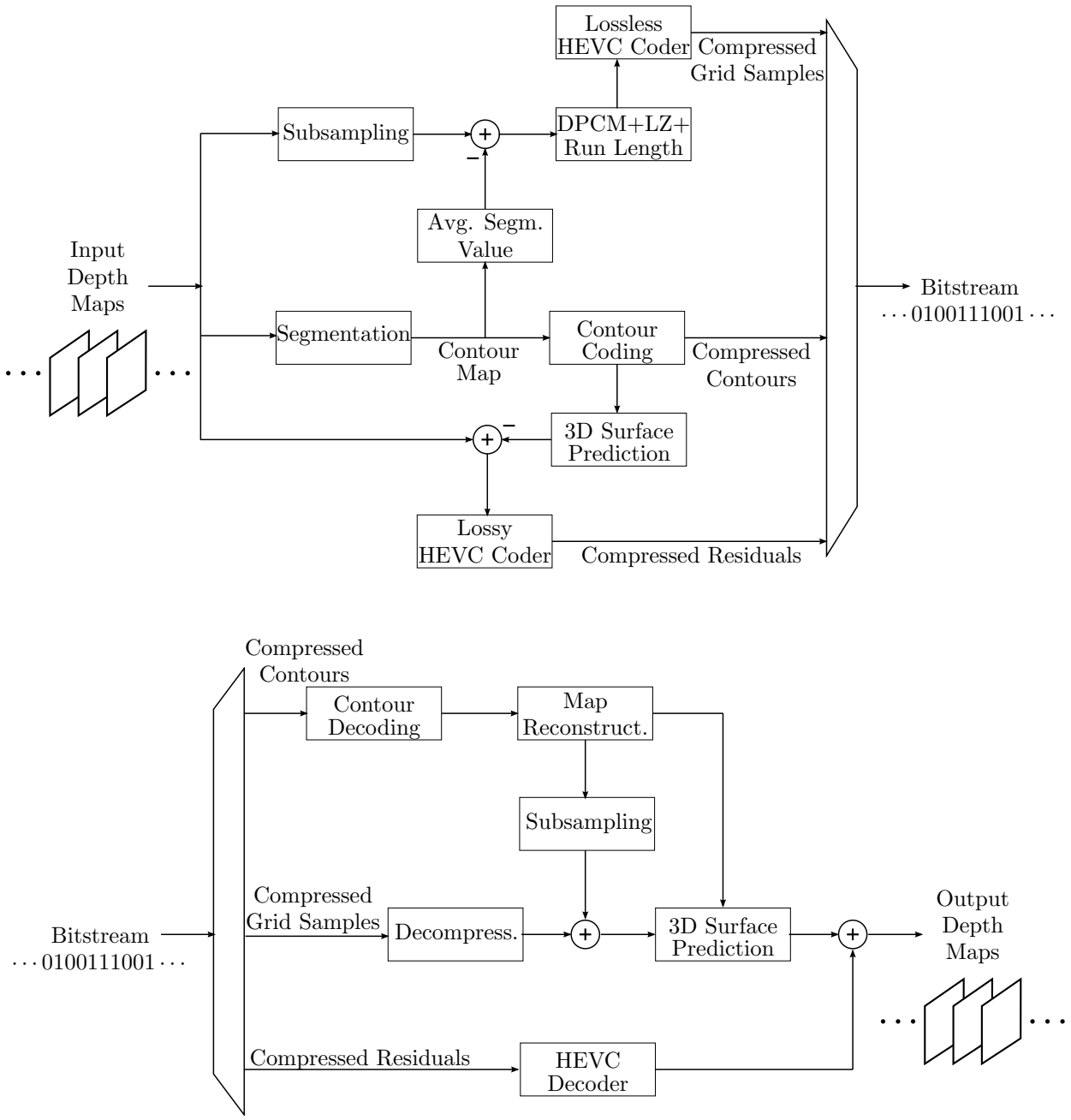


Figure 3.6: Coder and decoder schemes for the depth map coding technique EC + 3D surface prediction.

which permits obtaining good coding gains with a limited complexity. For the following frames, a DPCM strategy is tailored in order to exploit temporal redundancy. Depth samples are first predicted from the previous frames (zero-motion prediction), and the prediction residual is coded like depth samples of the Intra depth frame. We resort to this simple technique because due to the very low resolution of the input images the size of the produced files is minimal.

The high resolution contour information and the low resolution depth data can be used to produce a very accurate prediction of the input high resolution depth map. For this step we used an approach derived from [ZC09] that here we briefly report. The key idea is that the depth map is made of a set of smooth surfaces represented by the segmented regions. For each region a set of samples is available, i.e., the subsampled depth map pixels belonging to that region. Each pixel p_i of the high resolution depth map is thus surrounded by 4 samples of the grid (see Fig. 3.8) and the idea is to predict it by interpolating only the grid samples belonging to the same region. If all the 4 samples belong to the same region of p_i, the estimation of p_i is simply given by the bilinear interpolation of the 4 samples. When instead the sample is close to the contour of one of the segmented regions, some of the 4 samples could belong to different regions and the value of p_i is estimated by interpolating only the values of the grid samples that belong to the same region. Up to the symmetry and excluding the trivial case of samples that correspond to grid points, there are 5 possible cases (see Fig. 3.8).

a) If all the 4 samples are inside the same region p_i is simply given by the bilinear interpolation of the 4 samples.

b) If p_i is surrounded by 3 samples belonging to the same region it is estimated by assuming that it lies on the plane defined by the 3 points.

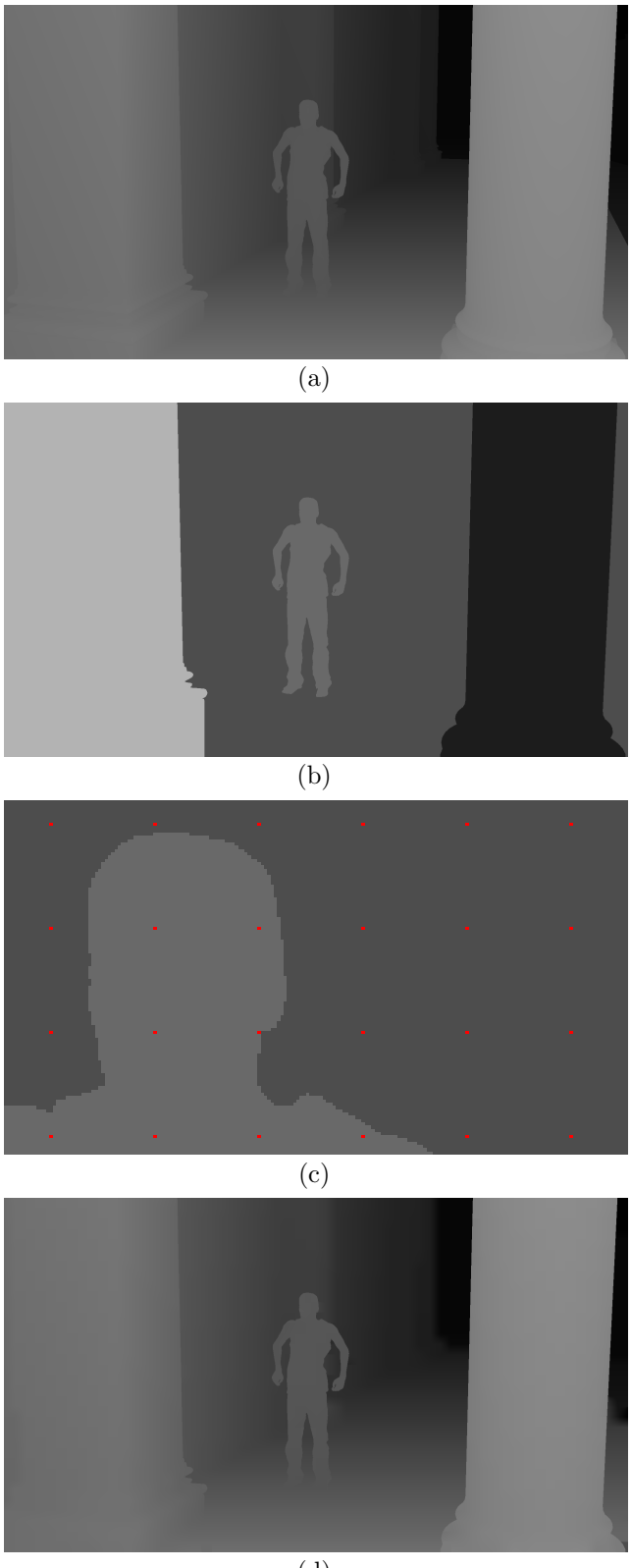
c) If p_i is surrounded by 2 neighbors in the region and 2 outside (e.g., when it is close to an edge) a prediction of the two points outside of the region is performed by assuming that each of them lies on the line passing through the closest available point and the symmetrical point with respect to the available one (the orange points in Fig. 3.8c). These points are used to compute p_i by bilinear interpolation.

d) If p_i has just one neighbor in the same region the value of this sample is taken as estimate.

e) If p_i has no neighbors the average depth value of the region is used.

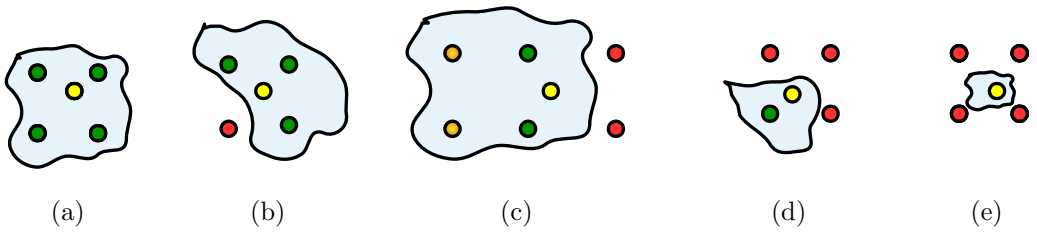
This approach allows to obtain a very accurate prediction of the input depth map, with only some small artifacts typically on the edges not captured by the segmentation.

Finally, the residual difference between the estimated and actual depth map is computed and lossy compressed using the HEVC coding engine. In this case, the main REext profile is used, enabling rate-distortion optimization.



(a) (b) (c) (d)

Figure 3.7: (a) Sample depth map from the *dancer* sequence; (b) segmented depth map; (c) detail of the segmented depth map with the subsampling grid (red dots represent the position of the grid samples); (d) depth map predicted from the contour and the low resolution samples.



(a) (b) (c) (d) (e)

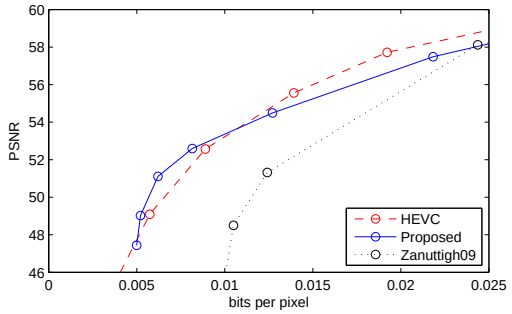
Figure 3.8: Grid samples: the 5 possible configurations. The unknown yellow pixel is estimated by using only the green pixels (plus the orange ones in case c).

3.3.2 Experimental setup

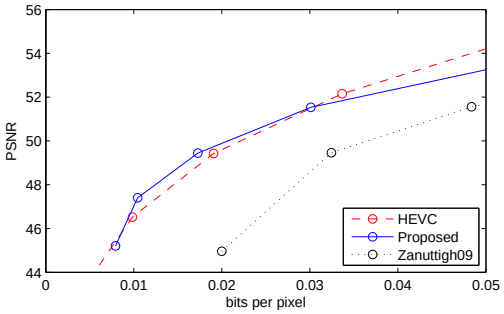
The performance of the proposed approach was evaluated on three different sequences with different resolutions and characteristics. The first is the *dancer* sequence, an high resolution (1920×1088) synthetic scene, the second is the *lovebird* sequence, a real world sequence with a resolution of 1024×768 and the third is the *mobile* sequence, another synthetic scene with resolution 720×540. We compared the proposed approach with the HEVC standard video coder (Intra mode, main profile) and with the segmentation-based depth coding approach of [ZC09]. Sampling grids of 16×16 (sequences *lovebird* and *mobile*) or 32×32 (sequence *dancer*) were used.

3.3.3 Results

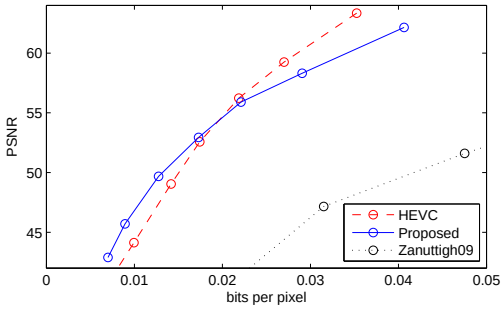
Figure 3.9(a) shows the performance of the proposed and competing approaches on the *dancer* sequence. There is a large performance gain (up to 4 dB) with respect to [ZC09] in the considered bit-rate range. The two approaches share the idea of exploiting segmentation and low resolution approximation, but the contour coding strategy of this work is more efficient than the arithmetic coder of [ZC09] and the proposed low resolution samples and residual compression strategies largely outperform the JPEG2000 based coding used in that work. The comparison with HEVC is more though: at low and medium bit-rates (up to 0.01 bpp) the proposed approach is able to outperform HEVC thanks to the efficient representation of the contours and to the very low information content of the residuals. The performance gain reaches around 2 dB at around 0.006 bpp. Notice how the contours remain sharp and not blurred on the whole bit-rate range while HEVC achieves this result only at high bit-rates. Figure 3.9(b) shows the results for the *lovebird* sequence: even if the resolution is different and the scene is a real world one (not synthetic), the results are very similar to the previous one, with the proposed approach able to outperform [ZC09] at all bitrates and HEVC for bitrates up to 0.02 bpp corresponding to around 53 dB. The *mobile* sequence (Figure 3.9(c)) is more challenging for our approach since the edges are already blurred in the input depth maps. This reduces the effectiveness of coding strategies based on the assumption of sharp edges between the various regions like the proposed one



(a) *dancer*



(b) *lovebird*



(c) *mobile*

Figure 3.9: Comparison of the performances of the proposed approach with the HEVC coder and with the method of Zanuttigh *et al.* [ZC09]. The test sequences are *dancer*, *lovebird*, and *mobile*.

and [ZC09] (that on this sequence has very poor performances). However our approach is still able to outperform HEVC at low bitrates. The maximum gain is around 3 dB in the bit-rate that goes from the minimum to 0.022 bpp.

The compressed depth maps are used in conjunction with the DIBR engine to synthesize novel views. The synthesized images obtained with depth maps compressed with the proposed technique are compared to the synthesized images obtained with depth maps compressed with HEVC Intra. In order to take into account the artifacts introduced by the DIBR software, the reference for this kind of images are the synthesized images obtained with uncompressed depth maps [EYUHG10]. In order to assess the quality of the virtual views, we compared them to synthesized images obtained by applying the same DIBR algorithm to the uncompressed depths and views. The comparison can be done either visually or by objective metrics.

From a subjective point of view the edge preserving capabilities of the proposed approach are particularly influential when the depth is used for view warping and interpolation. At the same time the proposed algorithm can introduce non-perceptible or non-annoying artifacts and objective metrics can assign low scores for them even if for a human observer the degradation is relatively acceptable.

Figure 3.10 shows a synthesized view of the *lovebird* sequence. Depth data compressed at around 0.008 bpp with both HEVC Intra and the proposed approach. From the figure it is clear how depth compressed with the proposed approach leads to a better interpolation, in particular notice how the regions close to the people edges have much smaller artifacts.

A frame from views 1 and 5 of the *dancer* sequence have been used to reconstruct view 3. Depth data compressed at around 0.005 bpp with both HEVC Intra and the proposed approach. In Figure 3.11 the synthesized images are reported. Again HEVC coding leads to more edge artifacts in proximity of the borders.

Figure 3.12 shows the PSNR of the synthesized images obtained using the depth maps compressed with the proposed technique and with HEVC Intra, while in Figure 3.13 is shown the SSIM. For the sequences *dancer* and *lovebird* the synthesized images produced with depth maps compressed with HEVC have a higher PSNR and SSIM in the considerate bit-rate range, while at low bit-rates (up to 0.017 bpp) the proposed approach has an advantage (up to 3 dB at the minimum bit-rate).

(a1) uncompressed

(a2) uncompressed, detail

(b1) HEVC Intra

(b2) HEVC Intra, detail

(c1) EC + 3D surface prediction

(c2) EC + 3D surface prediction, detail

Figure 3.10: Sequence *lovebird*: different artifacts introduced by HEVC Intra (b) and the proposed technique EC + 3D prediction (c). The reference image (a), obtained using uncompressed depth maps, is reported to show the artifacts introduced by the rendering. HEVC Intra and the technique EC + 3D prediction are used at around 0.008 bits per pixel.

(a1) uncompressed

(a2) uncompressed, detail

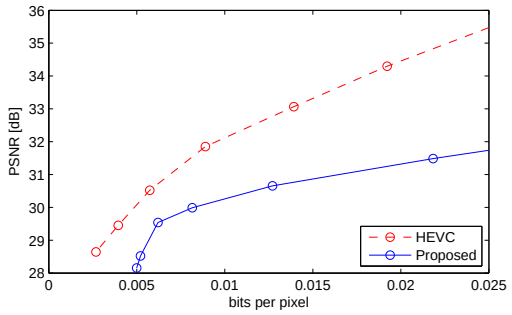
(b1) HEVC Intra

(b2) HEVC Intra, detail

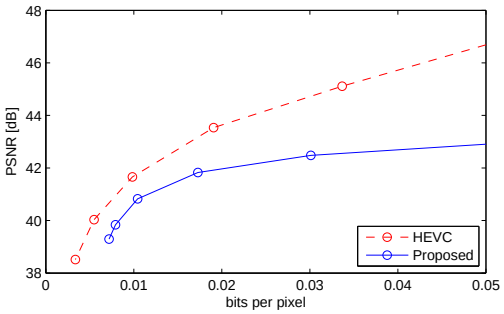
(c1) EC + 3D surface prediction

(c2) EC + 3D surface prediction, detail

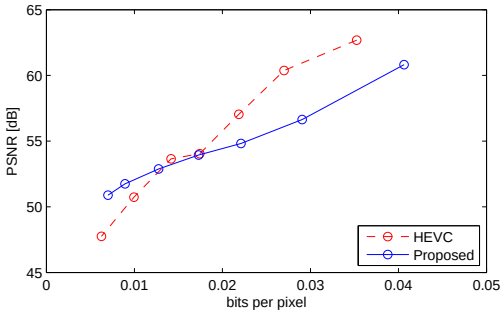
Figure 3.11: Sequence *dancer*: different artifacts introduced by HEVC Intra (b) and the proposed technique EC + 3D prediction (c). The reference image (a), obtained using uncompressed depth maps, is reported to show the artifacts introduced by the rendering. HEVC Intra and the technique EC + 3D prediction are used at around 0.005 bits per pixel.



(a) *dancer*

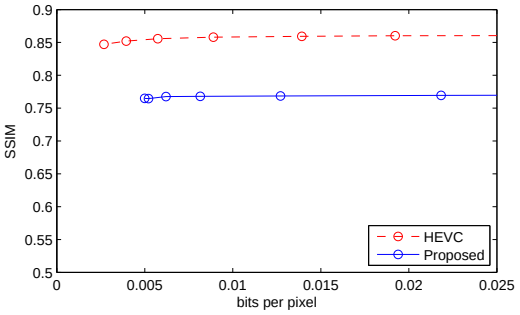


(b) *lovebird*

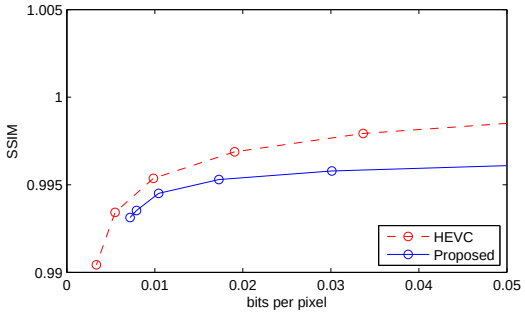


(c) *mobile*

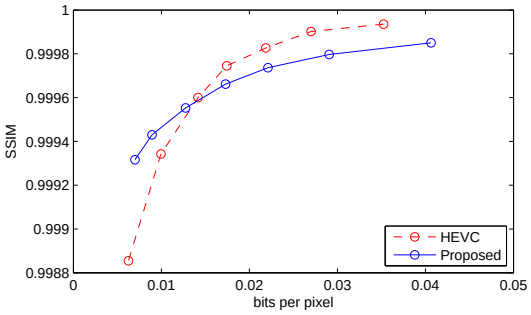
Figure 3.12: Depth map coding technique EC + 3D surface prediction: PSNR of the synthesized images obtained using the compressed depth maps for the sequences *dancer* (a), *lovebird* (b), and *mobile* (c).



(a) *dancer*



(b) *lovebird*



(c) *mobile*

Figure 3.13: Depth map coding technique EC + 3D surface prediction: SSIM of the synthesized images obtained using the compressed depth maps for the sequences *dancer* (a), *lovebird* (b), and *mobile* (c).

3.4 Conclusions

This chapter focuses on segmentation-based coding of depth maps, introducing the concepts of elastic deformation of curves in the context of depth map coding. Two depth map coding techniques have been discussed. Both of them use the lossless contour coding technique discussed in Chapter 2 to describe the object contours of a segmented scene, while the approaches to represent the inner part of the objects are on SA-SPIHT for the first technique, and a prediction from a subsampled version of the original depth map for the second one.

The results obtained with the proposed techniques show that with smart contour coding even segmented coding approaches are able to compete with HEVC Intra. Moreover the preservation of the contour information allows a very high quality synthesis of novel views.

The technique EC + SA-SPIHT presented in this chapter is published in the journal article [CCPP14], while the technique EC + 3D surface prediction is published in [CZM+16].

Chapter 4

Contour-based depth coding: a subjective quality assessment study

Contents

The video-plus-depth representation for multi-view video sequences (MVD) consists of several views of the same scene with their associated depth information, which is the distance from the camera for every point in the view [DPPC13]. Depth information allows synthesizing virtual view points, for such applications as 3D television and free-viewpoint

video, but it requires ad-hoc compression techniques, since those developed for texture images are not well suited for depth maps.

A key point in depth image compression is that depth maps are not meant to be visualized but only used for rendering of virtual views. Various techniques have been specifically proposed to code the depth information associated to the views and recent approaches include contour-based and object-based coding of depths, as we have seen in Section 3.1. This approach seems reasonable because the properties of depth maps differ greatly from the ones of texture images. Objects within a depth map are usually arranged along planes in different perspectives. As a consequence there are areas of smoothly varying levels, separated by sharp edges which correspond to object boundaries. It is generally recognized that a high-quality view rendering at the receiver side is possible only by preserving the contour information [GLG12], [DCF12], [SKN+10], since distortions on edges during the encoding step would cause a sensible degradation on the synthesized view and on the 3D perception. In case the edges of the depth map are losslessly compressed and the inner part is coarsely compressed, the typical artifact consists in a slight displacement of an object, with clear sharp edges. This leads to low scores for objective metrics like PSNR, which are very sensitive to this kind of errors, while it should leave a low impact on the perceived quality. To the best of our knowledge no study has been done so far to validate this claim [BPLC+11a] [BKP+11] [BPLC+12] [DJC+15], so we conducted a subjective test to assess the quality of synthesized images, obtained using DIBR software and depth maps compressed either with an object-based technique, or a hybrid block-based technique.

In Chapter 2 a lossless contour coding technique that uses elastic deformation of curves to losslessly encode the object contours is described, while in Section 3.2.1 is described a technique that uses the shape-adaptive wavelet transform to represent the inner parts of the objects. In a practical bit-rate range the combination of these two techniques proved to be competitive, in terms of objective quality metrics (PSNR and SSIM), with HEVC [SOHW12], state-of-the-art technique for hybrid block-based coding. Just like other contour-based methods, the technique EC + SA-SPIHT presents totally different artifacts in synthesized images with respect to HEVC, and their impact is difficult to evaluate with classical metrics like the PSNR.

The results of the subjective test show that the contour information is indeed relevant in the synthesis step: preserving the contours and coding coarsely the rest typically leads to images that users cannot tell apart from the reference ones, even at low bit rate. Moreover, our results show that objective metrics that are commonly used to evaluate synthesized images may have a low correlation coefficient with MOS rates and are in general not consistent across several techniques and contents.

The rest of the chapter is organized as follows: in Section 4.1 the basics of subjective visual quality assessment are given; in Section 4.2 an overview of the coding techniques used for the preparation of the test is presented; the Section 4.3 deals with the test design,

the participants, the test environment, the used stimuli and the test procedure; finally the Section 4.4 contains the results, followed by the conclusions.

4.1 Background notions on quality assessment

The evaluation of lossy compression techniques for videos and images can be conducted through objective or subjective methods [Ric04]. Objective metrics are simply computed from the signals' values and they can be distinguished into perceptual and non-perceptual metrics. Subjective methods on the other hand ask a group of people to judge the quality of the signal.

The most common objective metrics for video quality evaluation are described in Section 1.1.4.

4.1.1 Subjective quality assessment tests

In a subjective visual quality assessment test, the viewers, usually called "*participants*", provide a set of responses to the received "*stimuli*", the images or video sequences. To consider these responses as a reliable measure of perceptual impressions, the experimenter must control a set of external conditions. The experimenter in particular has the choice of the stimuli, as well as their order and repetitions; the number and the characteristics of the subjects; the environment for the experiment, as well as its methodology, which includes the question that the subject has to answer, along with the rating scale. The variables that the experimenter controls are called *independent variables*.

The response of a subject to a stimulus is the *dependent variable*, and the whole test is designed so that the variation in the subjective responses is caused only by the prompted stimuli.

The experimenter formulates a question to be answered in order to confirm or reject the *null hypothesis*, which indicates how likely it is that the research hypothesis is true or false. Example of null hypothesis is that the images compressed with technique "A" and the images compressed with technique "B" have the same quality at equal bit-rate.

As only a limited number of people can participate to a subjective test, the experiment has to be correctly designed to generalize the confirmed or rejected hypothesis [BZ07]. In order to verify whether a test is correctly designed we can check the two properties of objectivity and reproducibility. Objectivity property refers to the fact that if we perform another test in another laboratory, with another test method and other subjects but using the same stimuli, we should get statistically similar results. While if for our test the reproducibility property is verified, a test that uses the same stimuli and test methodology, in another laboratory, with different people, produces statistically comparable data. To obtain objective and reproducible results, all the external conditions and the research question must be carefully determined and the collected results reported in all the details

[Win05] [BHH78] [SC67].

4.1.2 Design of a subjective test

The aim of the experiment defines the research question, which can be focused either on a specific characteristic of the stimuli, or on the overall impression that the subjects receive from the stimuli [Int05].

An image or a video is a complex signal, and when such a stimulus is presented to a subject it is likely that he perceives multiple attributes (such as sharpness, contrast, etc.) at the same time. A *perceptual measurement* aims to measure a specific attribute of the stimuli, thus the question to be asked to the participants will not be about the overall perceived quality of the pictures or video sequences, but just about the specific attribute under study, with a detailed and clear formulation. To measure different attributes, it is recommended to evaluate them separately.

Unlike perceptual measurements, the *affective measurement* has the purpose of quantifying the general impression of the stimulus, such as acceptance, liking, or annoyance. All the attributes, the environment and the context, the state of mind of the subject, are all blended into a single impression. A short training is needed to explain the experimental design and the research question. Three classes of affective measurement can be defined: acceptance tests, where the subject is asked to report the degree of liking of the stimulus; appropriateness tests, where the degree of liking is contextualized; preference tests, where a preference choice has to be made between different stimuli.

Stimuli and subjects

The test material has to be representative of the data used in a realistic scenario and at the same time its characteristics should be challenging for the attributes that the experimenter wants to take into analysis [ITU12b]. The number of stimuli to be used in a test is variable and depends on the factors that the experimenter is considering. Ideally every combination of treatments should be showed to the subjects (full factorial design), but this approach leads to a very large number of stimuli. Given that every session should not last more than 30 minutes, a fractional factorial design is preferred: the stimuli set contains a just a significant fraction of all possible stimuli [BHH78].

As the test material has to be relevant for a use in a realistic scenario, the same is true for the subject who participate in the test. We can distinguish two categories: expert and untrained (or naive). Concerning their area of expertise, the experts represent a sensitive part of the total population and they take part in a subjective test when the experimenter wants a detailed evaluation of the stimuli. On the other hand for affective tests the untrained subjects are preferred, as they reproduce a random sampling of the population [DS12] [ITU12b]. Depending on the nature of the study, naïve or experienced participants are used for the assessment.

Experimental set-up

In order to perform a meaningful test for quality assessment, the experimental set-up should recreate the usage conditions for the targeted application. At the same time all the external experimental parameters that could bias the impression of the subjects must be restrained.

A standardization effort has been done in order to facilitate the test reproducibility. Recommendations and set-up parameters for the reproduction system and the environment where the test is performed are indicated in [ITU12b] and [ITU12a]. However, as the technology evolved very rapidly over the last years, the experimenter can define an ad-hoc configuration to study a particular application, as long as he provides a detailed description of the test environment in his report.

Test methods

Once decided the target application and the aim of the test, the experimenter has to choose a test method: the question to be asked to the subjects, and how the subjects have to answer. Rating scales are used to translate the impression perceived by the subject from a stimulus into a measure.

Scaling methods are either direct or indirect. In *direct scaling* methods the overall impression perceived by the subject is directly translated into a category or a number, proportional to the magnitude of the sensation caused by the stimulus [Ste46]. Regarding *indirect scaling* methods, they are used to measure the perceived impression by comparing the effects of two stimuli: the more they are different the more the subject will perceive different sensations. By indicating the stimulus that is considered better than the other, if they are not perceived as equal, the experimenter can construct a matrix of probabilities that can in turn be converted into scores on interval scales [BZ07].

Among common test methods we can mention double stimulus, single stimulus, and stimulus comparison methods, all accurately described in the ITU recommendations [ITU12b], [ITU07] and [ITU99]. *Double stimulus* methods belong to the direct scaling methods family. Two stimuli are presented to the subject, at the same time or sequentially, one of the two being the reference and the other the test stimulus. The subject has then to evaluate both, or just the test stimulus, to measure the fidelity of the test stimulus to a reference one.

A common implementation of double stimulus methods is the *DSIS* test (*Double Stimulus Impairment Scale*): the viewer is asked to evaluate only the test stimuli using a rating scale composed of five intervals (ITU-R five-grade discrete impairment scale [ITU12b]). An example of the scale is reported in Figure 4.1, and it can be discrete or continuous.

In *single stimulus* methods there is no reference and only one stimulus is shown at the time. The user will have to choose the degree of acceptance of the stimulus on a discrete or continuous scale. This kind of methods have the advantage of being closer to real-world

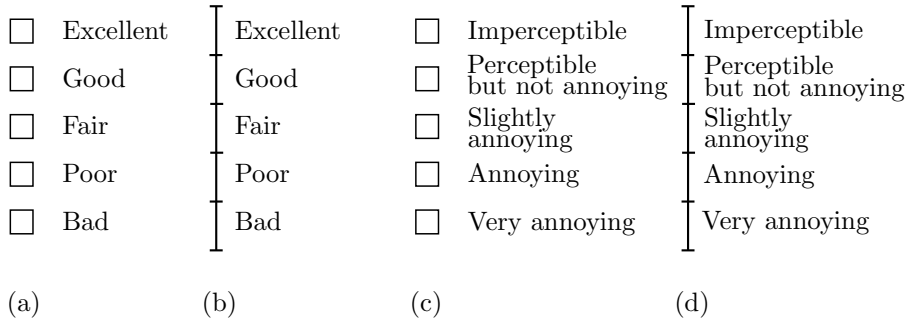


Figure 4.1: Rating scales: ITU-R 5 grade discrete (a) and continuous (b) quality interval scale, ITU-R 5 grade discrete (c) and continuous (d) impairment interval scale.

application than double stimulus methods, but at the same time small differences may not be perceived.

On the other hand, *stimulus comparison* methods allow to precisely discriminate different stimuli. Pairwise-comparison methods are considerate the most precise: stimuli are presented in pairs and the subject has to use a scale to establish a better-worse relation between the two. The problem related to this approach is that the sessions tend to be very long.

4.1.3 Analysis of subjective results

Once the results have been collected, a pre-process is needed to detect and potentially remove the outliers. The data can then be analysed using appropriate statistics, depending on the experiment design and the nature of the study. Common statistics, as reported in [ITU12b] and [ITU04], are Spearman's correlation coefficient, root mean square error and mean opinion score.

Outlier detection and removal

The data the experimenter deals with is a collection of scores. There are many ways to perform the outlier detection, a common one is that if the experimenter cannot make any assumption on the score distribution, a score x is considered as an outlier if its value is out of the interquartile range by a distance of more 1.5 times the interquartile range.

$$x \text{ is outlier:} \quad x < \mathrm{Q1} - 1.5 \cdot \mathrm{IQR} \quad | \quad x > \mathrm{Q3} + 1.5 \cdot \mathrm{IQR} \tag{4.1}$$

where $\mathrm{IQR} = \mathrm{Q3} - \mathrm{Q1}$, and Q1 and Q3 are the first and third quartile, respectively. Then, if the number of outliers among the scores of a subject is greater than the threshold of 10% or 20%, he is labelled as an outlier and all of his scores are discarded [DSGLE11] [DS12].

Mean opinion score

To summarize the collected data the experimenter can compute and report the value of descriptive statistics like mean, standard deviation or variance, as well as the box-plot diagram, which graphically describe the score distribution through its quartiles.

The opinions of the subjects are also averaged into the *Mean Opinion Score* (MOS) to provide a concise representation of the appreciation for each stimulus. Given a stimulus j, its MOS can be calculated as:

$$\mathrm{MOS}_j = \frac{1}{N}\sum_{i=n}^{N} x_{nj}, \tag{4.2}$$

where N is the number of subjects left after the outlier removal, and x_{nj} is the rating given by subject n to the stimulus j [ITU12b].

To measure the general agreement between the participants, MOS values should always be reported with their *confidence interval*. Since the number of participants is usually small compared to the whole population, the CI must be calculated with the Student's t-distribution.

$$\mathrm{CI}_j = t_{\alpha,N-1}\frac{s_j}{\sqrt{N}}, \tag{4.3}$$

where $t_{\alpha,N-1}$ is the t-value corresponding to a two-tailed t-student distribution with $N-1$ degrees of freedom (the number of independent observations in the set of data) and s_j is the standard deviation of the scores assigned to the stimulus j. The significance level α for the $100 \times (1-\alpha)\%$ CI is usually set to 0.05, i.e. a degree of confidence of 95%.

4.2 Depth map coding techniques

To produce the test material for our analysis we will compress the depth maps using three different techniques: HEVC, the state-of-the-art hybrid block-based technique, the technique described in Section 3.2.1, and a simple control technique.

4.2.1 Depth map coding with elastic deformation of contours and SA-SPIHT

The technique described in Section 3.2.1 uses the elastic prediction as a context for an arithmetic coder, to improve the probability distribution for each symbol that composes the curve. The lossless coding of the contour is performed through an arithmetic coder, and the input symbol probability distribution is modified on the fly according to the elastic prediction, as we have seen in Chapter 2.

A segmentation map of the scene with different objects can be coded with our lossless contour coding technique, and this map can be used in conjunction with an object-based coding technique to code the depth images. The proposed method relies on the SA Wavelet Transform [LL00], followed by SA SPIHT (Set Partitioning In Hierarchical Trees) [CPVZ04],

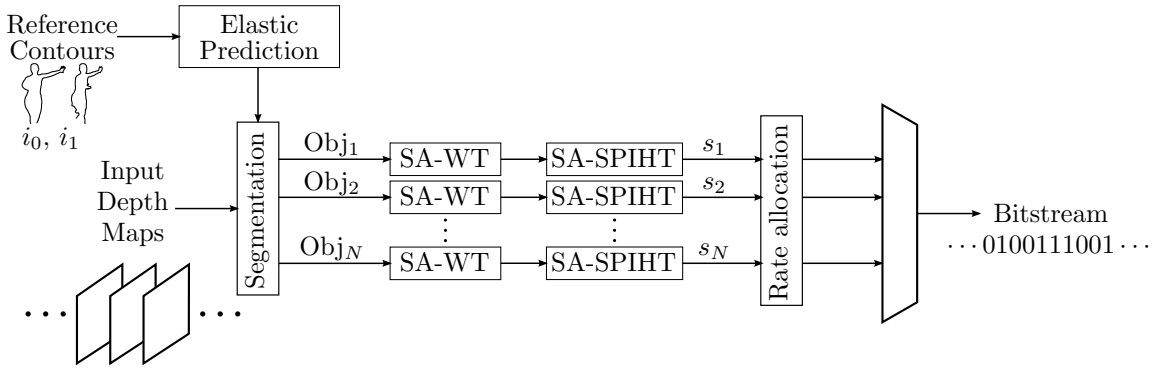


Figure 4.2: Coding scheme for the technique NR (No Refinement).

followed by an arithmetic coder for the SPIHT symbols (memoryless, without context). This provides an entirely Intra technique for the inner part of the objects. From now on we will refer to this technique as "CC", contour coding.

4.2.2 Advantages of lossless coding - Simple control technique

The validity of the object-based approach is suggested by the relevance of the contour information in the synthesis step. A minimum bit budget is needed for a *lossless* representation of contours and this initial cost may be a relevant fraction of the total rate dedicated to the depth map. One question that arises is whether the benefits of a lossless contour will actually be perceived as relevant against its bit-rate cost. We investigated on this subject by simply skipping the lossless coding part in our coding technique and using the elastic prediction to directly generate the segmentation map to feed the object-based coder. We rely on the SA SPIHT block to correct the possible imprecision of the coded depth map.

This control technique allows the rates to be lowered dramatically, however the quality of the synthesized images obtained from the depths compressed with this simple technique must be carefully evaluated. From now on we will refer to this technique as "NR", no refinement.

A concise scheme of the NR method is shown in Fig. 4.2. As inputs we have a depth image and the two reference contours needed for the elastic interpolation. First the elastic prediction is performed and the generated contours are used to arrange a segmentation map. The resulting objects are coded with a SA wavelet transform, followed by SA SPIHT.

4.3 Subjective test

4.3.1 Test design

The subjective evaluation has been performed following the Double Stimulus Impairment Scale (DSIS) methodology [ITU12b]. For each "round" a pair of images has been proposed

content	frames	view L	goal	view R	crop
beergarden	54-58	5	5.25	6	840×896
lovebird	1-5	6	7	8	1024×768
mobile	43-47	3	3.75	4	720×536

Table 4.1: Frames used for compression and synthesis for each sequence.

to the user, stimuli A and B, in which the stimulus A was always the reference, and stimulus B was the image to be evaluated. The reference image is obtained using uncompressed depths in the synthesis step, while the image to be evaluated is obtained with compressed depths. The viewer was informed of the presence of the reference image in the pair, and was asked to rate the drop of quality of the second image with respect to the first one, using a continuous scale ranging from 0 to 100, in which five ranges were associated with five distinct adjectives ("Very annoying", "Annoying", "Slightly annoying", "Perceptible", "Imperceptible").

4.3.2 Participants

A panel of twenty people took part in our test, 6 female and 14 male, aged from 23 to 32, with an average of 27.65 years. The subjects reported normal or corrected-to-normal visual acuity.

4.3.3 Test environment

The images were displayed on a DELL P2210 screen at their original resolution (reported in Tab. 4.1). The test space was set up with mid gray non reflective background, and isolated from external sources of light, as recommended in [ITU12b], [ITU12a]. To avoid direct light sources in the field of view of the user, except for the screen, we placed a lamp at 6500K color temperature behind the screen to provide ambient illumination. The resulting ambient light measured in front of the screen, when this was off, was approximately of 10 cd/m^2. Viewers participated to test sessions one at the time, sitting in front of the screen at a distance of its diagonal approximately, which corresponds roughly to three times the height of the images used for the test. The angular resolution from that distance is about 35 pixels per degree.

4.3.4 Stimuli

The multiview sequences *beergarden* (provided by Philips), *lovebird* (ETRI/MPEG Korea Forum) and *mobile* (Philips) have been used for test. An example for each content is shown in Fig. 4.3. We produced the test material by synthesizing novel views from uncompressed texture videos and compressed depth maps For each sequence a target view and time instant is chosen. Depth maps are then compressed using the techniques CC, NR, and HEVC.

beergarden (1920×1080)

lovebird (1024×768)

mobile (720×536)

Figure 4.3: Contents of the different sequences and their resolution.

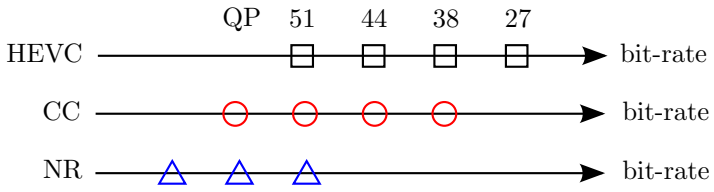


Figure 4.4: Coding rate scheme for the different techniques used in the test.

(a) (b)

Figure 4.5: Example of artifact introduced by the compression with the "no refinement" (NR) technique: imprecise contours will produce in the synthesized image a gap between the objects. The whole image is reported (a), along with a detail (b).

For HEVC, used in mode Intra, we have chosen as QPs: 27, 38, 44, 51. QP 51 is the maximum level of compression allowed, corresponding to the smallest bit-rate for HEVC. For CC, the use of SA-SPIHT allows to chose the exact bit-rate, so we have chosen the lowest possible rate and the bit-rates corresponding to QPs 51, 44, and 38. And as for NR, we have chosen again the lowest possible rate and other two to match the low rates of the two other techniques. A simple scheme on how the rates are selected for each technique is shown in Figure 4.4. For each sequence, 11 synthesized images are used in the test, leading to a total of 33 comparisons in the whole test. Details about the frames and views used for each sequence are reported in Tab. 4.1.

The use of the technique NR for this test is important to evaluate the effects of contour coding on the synthesized image. It will be shown that the precision of the contour leads to remarkable gains in terms of perceived quality. Moreover we used this technique at a very low bit rate as the reference for the "very annoying" quality, for the evident artifacts that it produces on the synthesized image. If the predicted contour lies outside the object and the interior part is coded very coarsely, we will produce a change in the geometry of the scene, that will result in a gap in between the objects of the synthesized image, as shown in Figure 4.5.

Once the depths are compressed for two views, we used them to generate new synthetic views in between the two with the Depth-Image-Based Rendering (DIBR) software [Feh04], as shown in Fig. 4.6. The use of synthesized images for the test is justified by the plausible scenario of Free Viewpoint Video (FVV) [BPLC+11b].

To perform the elastic prediction we need two reference curves, be them from two different views or two different time instants of the same view. These curves represent a sort of "intra" curves and they are coded independently from the others (and losslessly, for the techniques CC and NR). The gap or distance between the two reference curves along the

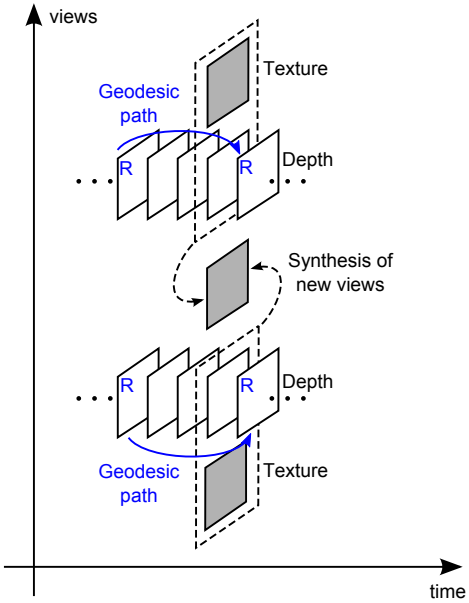


Figure 4.6: Elastic prediction and synthesis scheme.

view or the time axis affects the final rate: the accuracy of the prediction decreases with the distance, but at the same time a wider gap between the two reference curves allows a lower bit-rate. The distance of the reference curves for the elastic prediction has to be chosen for the techniques CC and NR. To better illustrate this concept, in Fig. 4.6 we marked with a blue R the frames used to extract a reference curve, a blue arrow representing the geodesic path from one to another, and the distance from a reference curve to the other is defined as the number of frames that exist in between the two. For both the techniques the chosen value is three frames in between the two reference ones and we perform the elastic prediction along the time axis. The synthesis is then performed using the compressed depths of two views and their associated texture images.

Regarding the reference images, the unavailability of intermediate views and the presence of evident synthesis artifacts lead us to the use of synthesized reference images, obtained with the original uncompressed depths, as suggested in [EYUHG10].

4.3.5 Procedure

Each image was shown for 7 seconds and preceded by a gray screen to indicate the stimulus (A or B) for 1.5 seconds. Every round was composed of: stimulus A, stimulus B, stimulus A, stimulus B, voting screen, as shown in Figure 4.7. The voting screen is a pop-up window on mid gray background. The voting window contains a continuous slider with the five adjectives on its side, and a text box that reports the rating value and again the correspondent adjective with a larger font size to be more evident, as shown in Fig. 4.8. The pairs of stimuli were presented in random order. The whole test took around 22 minutes to complete for each user.

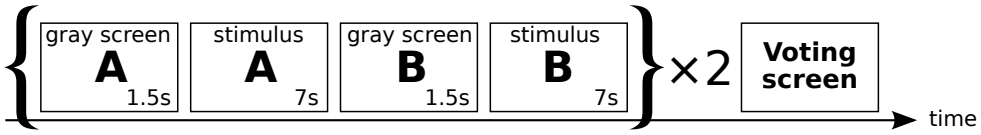


Figure 4.7: Procedure for each round of the test.

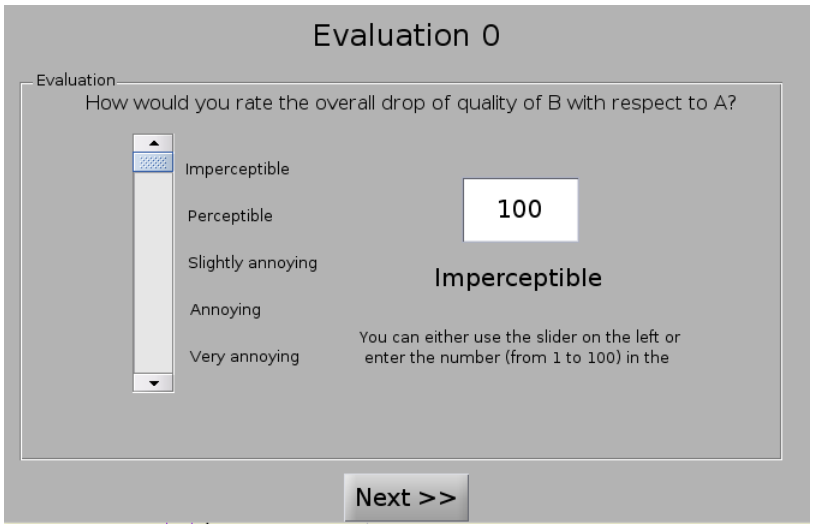


Figure 4.8: Voting window used in the test.

4.4 Results

The subjective scores were screened in order to detect and remove possible outliers, subjects whose scores differ greatly from the others'. We followed the procedure described in [ITU12b] for the DSIS test methodology. As we have not done any assumption on the distribution characteristics, values outside the interquartile range (from the 25% and the 75% percentiles) for more than 1.5 times are labeled outliers, and if there is an incidence of outliers in a subject's score of more than the 10%, he is considered an outlier and his scores are not taken in account. In our results three subjects have been marked as outliers and their scores have been discarded.

After the outlier removal, we verified that for each stimulus the score distribution is unimodal and we computed the Mean Opinion Scores (MOS), along with the 95% confidence interval (CI), with the assumption that the scores are following a t-Student distribution.

Objective measurements and perceived quality. In addition to the MOS, the images were also evaluated through different objective metrics. We considered pixel-based metrics: Peak Signal-to-Noise Ratio (PSNR) and Weighted Signal-to-Noise Ratio (WSNR) [MV93]; as well as non-pixel-based metrics: Multi-scale Structural Similarity (MSSIM) [WBSS04] and Visual Information Fidelity (VIF) [SB06]. In Figures 4.9, 4.10 and 4.11 the results of our subjective test and all the considered objective metrics are reported. Each column refers to a sequence, and each row to a metric. In particular, the computed MOS scores

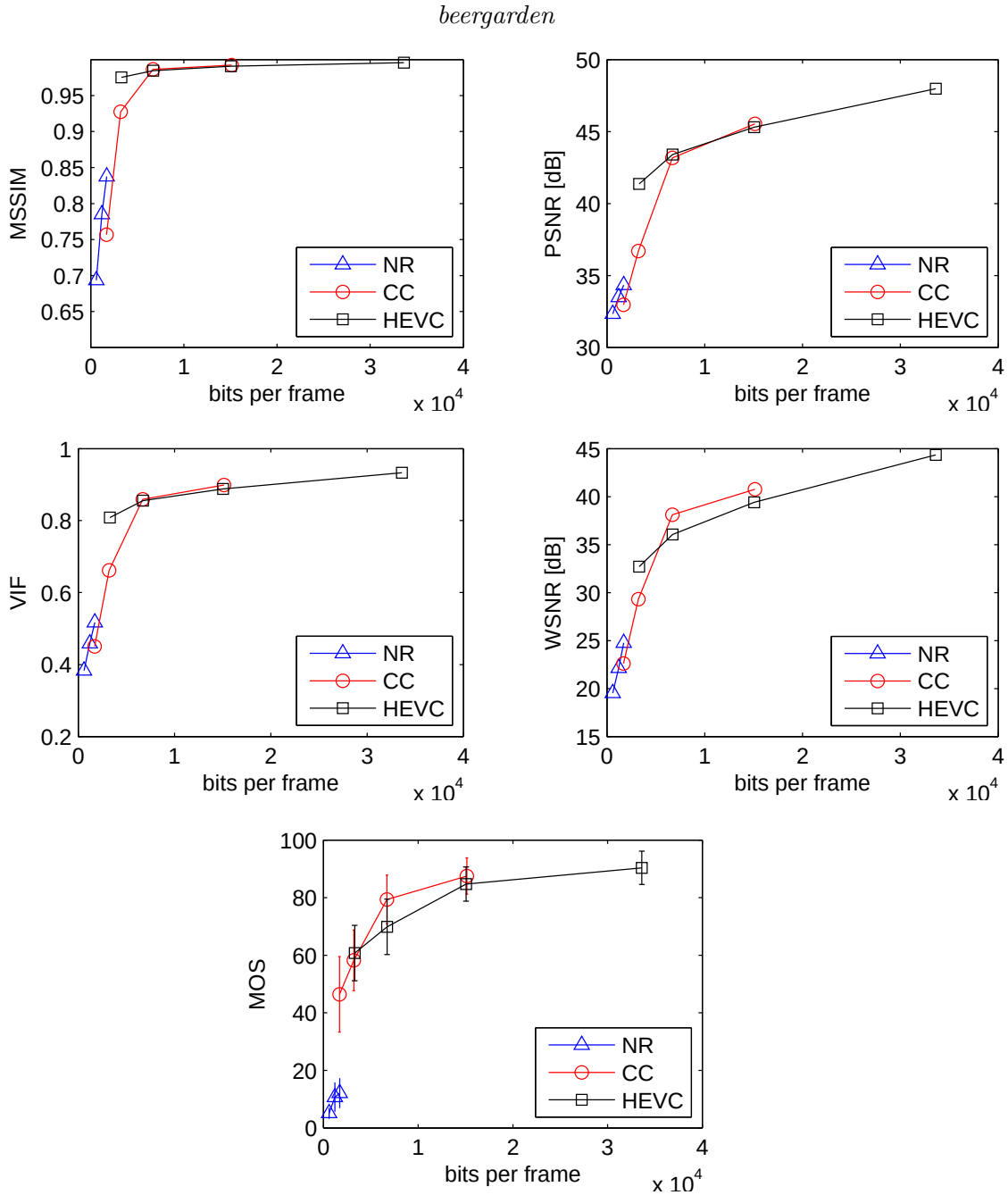


Figure 4.9: Sequence *beergarden*: mean opinion scores (MOS) with the 95% confidence interval and different objective metrics: Peak Signal-to-Noise Ratio (PSNR), Weighted Signal-to-Noise Ratio (WSNR), Multi-scale Structural Similarity (MSSIM), and Visual Information Fidelity (VIF). The techniques used are the simple control technique "no refinement" (NR), the contour coding technique (CC), and HEVC Intra (HEVC).

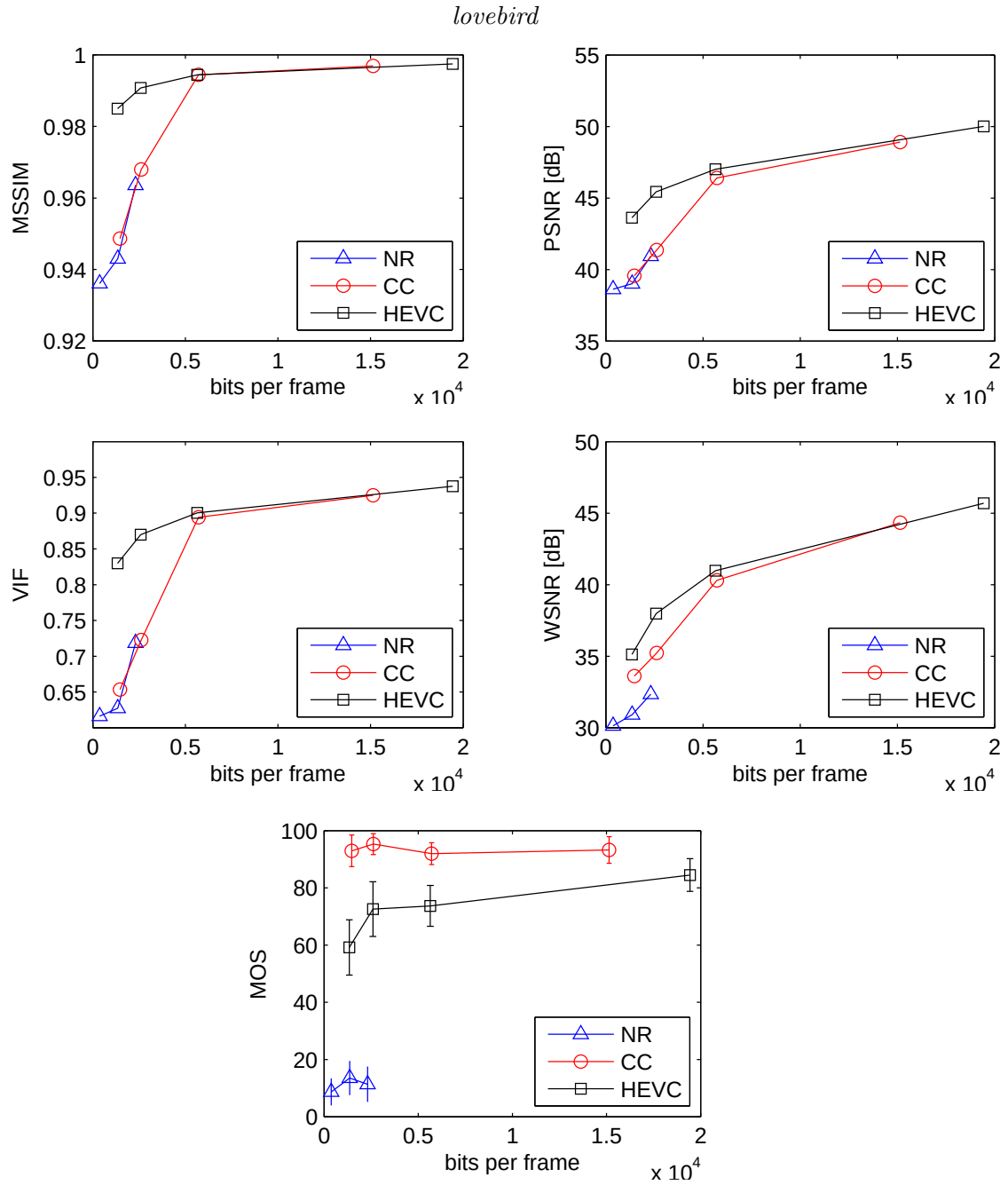


Figure 4.10: Sequence *lovebird*: mean opinion scores (MOS) with the 95% confidence interval and different objective metrics: Peak Signal-to-Noise Ratio (PSNR), Weighted Signal-to-Noise Ratio (WSNR), Multi-scale Structural Similarity (MSSIM), and Visual Information Fidelity (VIF). The techniques used are the simple control technique "no refinement" (NR), the contour coding technique (CC), and HEVC Intra (HEVC).

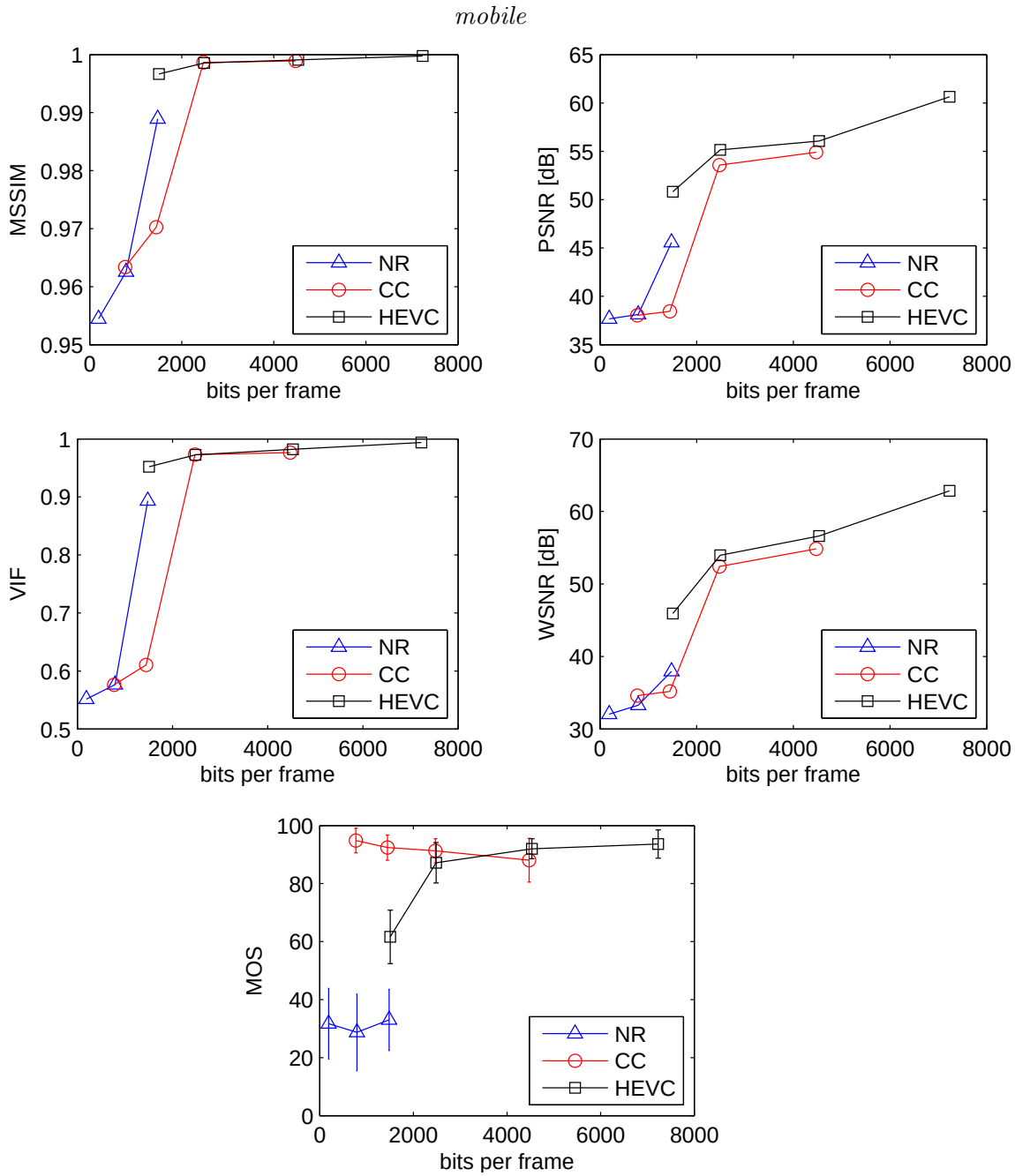


Figure 4.11: Sequence *mobile*: mean opinion scores (MOS) with the 95% confidence interval and different objective metrics: Peak Signal-to-Noise Ratio (PSNR), Weighted Signal-to-Noise Ratio (WSNR), Multi-scale Structural Similarity (MSSIM), and Visual Information Fidelity (VIF). The techniques used are the simple control technique "no refinement" (NR), the contour coding technique (CC), and HEVC Intra (HEVC).

$\lvert\rho\rvert$	overall	*beergarden*	*lovebird*	*mobile*	NC	CC	HEVC
MSSIM	0.62	0.97	0.52	0.46	0.75	0.13	0.80
VIF	0.55	0.97	0.50	0.37	0.52	0.02	0.79
PSNR	0.56	0.96	0.50	0.35	0.52	0.15	0.75
WSNR	0.68	0.99	0.63	0.45	0.80	0.14	0.79

p-value	overall	*beergarden*	*lovebird*	*mobile*	NC	CC	HEVC
MSSIM	0.00	0.00	0.11	0.15	0.03	0.68	0.00
VIF	0.00	0.00	0.12	0.26	0.16	0.96	0.00
PSNR	0.00	0.00	0.12	0.30	0.16	0.64	0.01
WSNR	0.00	0.00	0.04	0.17	0.01	0.67	0.00

Table 4.2: Spearman correlation coefficients ρ (in modulus) and p-values, calculated between the MOS and the objective metrics, for each content and technique.

with their 95% CI are reported in the first row.

As we can see in Figures 4.9, 4.10 and 4.11, HEVC exhibits very regular trends in its MOS scores. The resulting visible artifacts reported by the participants are mainly edge fragmentation and blocking effect, which disappear as the bit-rate increases. MOS scores of HEVC generally vary from the "perceptible" to the "imperceptible" range. The technique CC exhibits in general very high scores compared to the other techniques (most of its scores are in fact in the "Imperceptible" range), thus the coding of the contours is generally worth its cost for the perceived quality of the resulting synthesized images. This is especially true if we compare the MOS scores of CC to the ones of NR at the same bit-rate: imperfections in the contour can lead to annoying artifacts and make the final quality drop, even if they are diminished by a finer coding of the interior part. Moreover we see that at low bit-rates a perfect contour and a coarsely coded interior part tend to lead to an excellent perception of the synthesized image. In contrast to this assertion we find slightly lower scores at low bit-rates for the content *beergarden*, where the proximity of the objects to the camera demands a less coarse coding of the interior part to reach a very high level of perceived quality.

Dealing with objective metrics, again in Figures 4.9, 4.10 and 4.11, at low bit-rates HEVC generally shows better results. Pixel-based metrics like PSNR are in fact very sensitive to the most prominent artifact produced by the CC coding technique: the different geometry of the scene, given by the coarsely coded interior part, results in a slightly different disposition of the objects, with respect to the image synthesized from uncompressed depths. To a human observer it can be very difficult to tell apart the two images, but even the shifting of few pixels for an object or the background can cause a low PSNR. On the other hand, in terms of PSNR, the technique NR proves to be competitive with the technique CC. The explanation lies in the fact that very localized errors, especially if they affect the shape of the objects, are perceived as annoying by a human observer, and the tested objective metrics are not able to take this effect into account.

Apart from PSNR, also the more refined objective metrics WSNR, MSSIM and VIF produce scores that can differ greatly from the scores produced during the subjective test. The table 4.2 reports the Spearman correlation coefficients ρ (in modulus) and p-values, calculated between the MOS and the objective metrics. Each column reports either a content or a technique: when a content is indicated the coefficients are calculated considering all the techniques, conversely when a technique is indicated the coefficients are calculated considering all the contents. The metrics with the highest overall correlation coefficient are WSNR (0.68) and MSSIM (0.62), followed by the PSNR (0.56) and finally the VIF (0.55). While dealing with some of the techniques or some of the contents these objective metrics could provide a good subjective MOS prediction, but they are not consistent across several techniques and contents, in particular they showed a low correlation coefficient ρ and a high p-value for the method CC, indicating a very low correlation.

4.5 Conclusions

This chapter addresses the issue of evaluating the effects of contour-preserving compression techniques on synthesized images by means of a subjective test. We compared, with subjective results and objective metrics, HEVC, the technique proposed in [CCPP14], and the presented simple technique NR. The positive impact of contour preserving depth coding on the perceived quality of synthesized images was confirmed by our study. Moreover by the comparison of subjective results and objective metrics it is also clear that the quality assessment of synthesized images in MVD through objective metrics presents unresolved problems: algorithms can introduce non-perceptible or non-annoying artifacts and commonly used objective metrics can assign low scores for them even if for a human observer the degradation is relatively acceptable. Further experiments should be conducted to take into account also temporal artifacts and 3D video perception.

This work has been published in [CCPP15a], while the original simple control technique NR (described in Section 4.2.2) has been introduced in the short paper [CCPP15b].

Conclusion and future work

Objectives of the thesis

In the multiple-view video plus depth context, the goal of this thesis was to develop a framework for depth map compression that delivers lossless contour coding for segmented objects in the depth map, taking advantage of the temporal and inter-view redundancy of the MVD format.

Three main parts of the development can be distinguished: first an original technique aimed at lossless encoding of contours has been developed. The proposed coding scheme is based on elastic deformation of curves, which make possible to compactly represent the contours exploiting the temporal consistency in different frames.

The second part is the finalization of the framework with an effective technique to code the inner depth field of the objects, given the contours coded with the lossless contour coding technique. Two different codecs have been developed: a simple one that makes use of the shape-adaptive wavelet transform, followed by the shape adaptive version of SPIHT (set partitioning in hierarchical trees), and a more refined technique that uses a 3D surface prediction algorithm to estimate the depth field from the coded contours and a subsampled version of the data.

The third part is a quality assessment study that compares, by means of a subjective test, the synthesized images obtained using depth maps compressed with a hybrid-block-based coder (HEVC Intra) and depth maps compressed with the first of the proposed contour-preserving codecs.

Summary

Contour coding

A new technique aimed at lossless coding of object contours, based on the elastic deformation of curves, has been proposed. A continuous evolution of elastic deformations can be modelled between two reference contour curves and An elastically deformed version of the reference contours can be sent with a reduced coding cost to the decoder, to be used as side information to improve the lossless coding of the actual contour. To use the suitable

portion of the elastic curve as side information to code the current point, the two curves are coupled by a function obtained with dynamic time warping. An accurate estimation of the symbol probabilities is achieved "following" the elastic curve. The current curve, represented by a differential chain code, is then encoded with an arithmetic coder that takes as input also the vector of symbol probabilities. To perform the decoding, just a few parameters have to be sent to correctly estimate the symbol probabilities.

Experimental results on contours extracted from several multiview video sequences show remarkable gains with respect to the reference techniques and to the state of the art.

Depth map coding

In the context of segmentation-based coding of depth maps, two depth map coding techniques have been proposed, bringing the concepts of elastic deformation of curves in the context of depth map coding. Both the codecs use the lossless contour coding technique discussed in Chapter 2 to describe the object contours of a segmented scene, while the approaches to represent the inner part of the objects are based on the shape-adaptive wavelet transform followed by the shape-adaptive version of SPIHT for the first technique, and a 3D surface prediction from a subsampled version of the original depth map for the second one.

For the first technique we resorted to an existing object-based technique since it can immediately benefit from an improved contour coding method. Despite the simplicity of the approach the results are satisfying, due to the nature of the data we want to compress. This first technique is very effective when the objects have a distance from the camera that ranges from medium to high. In that case the coarse representation of the geometry of the inner part of the objects do not cause any problem in the synthesis of novel views because the depth region is rather smooth. On the other hand, when an object is very close to the camera, it can have a uneven depth region, which would require a large amount of bits to be coded with an acceptable quality for synthesis purposes.

We therefore developed a second coding scheme where the depth data is represented by a set of contours defining the various regions and a compact representation of the values inside each region. A 3D surface prediction algorithm is then used in order to obtain an accurate estimation of the depth field from the contours and a subsampled version of the data. The low resolution data and the prediction residuals are compressed with ad-hoc strategies.

The results obtained with the proposed techniques show that with smart contour coding even segmented coding approaches are able to compete with HEVC Intra. Moreover the preservation of the contour information allows a very high quality synthesis of novel views.

Quality assessment

It is generally recognized that a high-quality view rendering at the receiver side is possible only by preserving the contour information, since distortions on edges during the encoding step would cause a sensible degradation on the synthesized view and on the 3D perception. To the best of our knowledge no study had been done to validate this claim. A subjective test was then necessary to assess the quality of synthesized images, obtained using DIBR software and depth maps compressed either with an object-based technique, or a hybrid block-based technique.

Three techniques have been considered: HEVC Intra, the proposed technique that makes use of elastic deformation of contours and shape-adaptive wavelet transform and shape-adaptive SPIHT, and a simple control technique that uses just the elastic prediction for the contours, to evaluate the effects on the synthesis image of an inaccurate contour.

A double stimulus impairment scale test was set up, where the participants were asked to evaluate the drop of quality of a stimulus, with respect to the reference. The results of the subjective test show that the contour information is indeed relevant in the synthesis step: preserving the contours and coding coarsely the rest typically leads to a slight displacement of the objects, but from the user's perspective this kind of artefact is not noticeable. In most of the cases even at low bit-rates the participants could not tell apart the images synthesized with the depth maps compressed with the proposed technique from the reference ones.

Future works

At the time of finalizing this manuscript, several interesting perspectives can be proposed to expand the work of this thesis.

Contour coding

So far only a monodimensional elastic interpolation has been considered, but we expect a more precise estimation if we can take into account 4 or 8 reference curves from different views and different times, thus leading to further improvements of the technique.

Depth map coding

To improve the coding of the inner field of the object in a depth map, the 3D surface prediction algorithm can be used also without a proper segmentation: the prediction can be done if just the most relevant contours are extracted from the scene. This process should be faster than a segmentation, moreover the association of the contours in different views or time instants can also be done performing the elastic deformation and looking for the minimum distance in the curve space between two contours.

Quality assessment

So far only still images have been considered for the quality assessment. Further experiments should to be conducted to take into account also temporal artifacts and 3D video perception.

Moreover the research of an objective metric able to assess the quality of the synthesized images (and video) should be conducted: algorithms can introduce non-perceptible or non-annoying artifacts and most of the commonly used objective metrics could assign low scores for them even if for a human observer the degradation is relatively acceptable.

Publications

Journal articles

1. Marco Calemme, Marco Cagnazzo, and Beatrice Pesquet-Popescu, "Lossless contour coding using elastic curves in multiview video plus depth", *APSIPA Transactions on Signal and Information Processing*, volume 3, December 2014.

Conference papers

1. Marco Calemme, Marco Cagnazzo, and Beatrice Pesquet-Popescu, "Depth coding and perceived quality for 3D video", *Proceedings of QoMEX*, Costa Navarino, Greece, May 2015.

2. Marco Calemme, Marco Cagnazzo, and Beatrice Pesquet-Popescu, "Contour-based depth coding: a subjective quality assessment study", *Proceedings of 2015 IEEE International Symposium on Multimedia (ISM)*, Miami, Florida, USA, December 2015.

3. Marco Calemme, Pietro Zanuttigh, Simone Milani, Marco Cagnazzo, and Beatrice Pesquet-Popescu, "Depth maps coding with elastic contours and 3D surface prediction", *Proceedings of ICIP 2016*, Phoenix, Arizona, USA, September 2016.

Bibliography

[Abr63] N. Abramson, *Information theory and coding*, McGraw-Hill electronic sciences series, McGraw-Hill, 1963. *Cited in Sec.* 1.5

[AEDF+15] A. Abou-Elailah, F. Dufaux, J. Farah, M. Cagnazzo, A. Srivastava, and B. Pesquet-Popescu, "Fusion of global and local motion estimation using foreground objects for distributed video coding". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25 (6), pp. 973–987, June 2015. *Cited in Sec.* 2.1.3

[BE97] F. Bossen and T. Ebrahimi, "A simple and efficient binary shape coding technique based on bitmap representation", in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 4, pp. 3129–3132, IEEE, 1997. *Cited in Sec.* 1.4.2

[BHH78] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for experimenters : an introduction to design, data analysis, and model building*, Wiley series in probability and mathematical statistics, J. Wiley & Sons, New York, Chichester, Brisbane, 1978. *Cited in Sec.* 4.1.1, 4.1.2

[Bjo01] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", in *VCEG Meeting*, Austin, USA, Apr. 2001. *Cited in Sec.* 3.2.3

[BKP+11] E. Bosc, M. Köppel, R. Pépion, M. Pressigout, L. Morin, P. Ndjiki-Nya, and P. L. Callet, "Can 3d synthesized views be reliably assessed through usual subjective and objective evaluation protocols?", in *2011 18th IEEE International Conference on Image Processing*, pp. 2597–2600, Sept 2011. *Cited in Sec.* 4

[BL11] G. Bjontegaard and K. Lillevold, "Context-adaptive vlc video transform coefficients encoding/decoding methods and apparatuses", Apr. 5 2011, uS Patent 7,920,629. *Cited in Sec.* 1.5.3

[Bov00] A. Bovik, *Handbook of Image & Video Processing*, Elsevier Academic Press, 2000. *Cited in Sec.* 1.1, 1.1.2

[BPLC+11a] E. Bosc, R. Pépion, P. Le Callet, M. Köppel, P. Ndjiki-Nya, L. Morin, and M. Pressigout, "Perceived quality of dibr-based synthesized views", in *SPIE Optical Engineering+ Applications*, pp. 81350I–81350I, International Society for Optics and Photonics, 2011. *Cited in Sec.* 4

[BPLC+11b] E. Bosc, R. Pepion, P. Le Callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin, "Towards a new quality metric for 3-d synthesized view assessment". *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5 (7), pp. 1332–1343, 2011. *Cited in Sec.* 4.3.4

[BPLC+12] E. Bosc, R. Pépion, P. Le Callet, M. Pressigout, and L. Morin, "Reliability of 2d quality assessment methods for synthesized views evaluation in stereoscopic viewing conditions", in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, pp. 1–4, IEEE, 2012. *Cited in Sec.* 4

[BWS+07] P. Benzie, J. Watson, P. Surman, I. Rakkolainen, K. Hopf, H. Urey, V. Sainov, and C. von Kopylow, "A survey of 3dtv displays: Techniques and technologies". *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 17 (11), pp. 1647–1658, Nov. 2007. *Cited in Sec.* 1.3

[BZ07] S. Bech and N. Zacharov, *Perceptual audio evaluation-Theory, method and application*, John Wiley & Sons, 2007. *Cited in Sec.* 4.1.1, 4.1.2

[CAB10] M. Cagnazzo, M. Antonini, and M. Barlaud, "Mutual information-based context quantization". *Signal Processing: Image Communication (Elsevier Science)*, vol. 25 (1), pp. 64–74, Jan. 2010. *Cited in Sec.* 2.1.2

[Can86] J. Canny, "A computational approach to edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 679–698, 1986. *Cited in Sec.* 2.3

[CB09] Y.-M. Chen and I. V. Bajić, "Compressed-domain moving region segmentation with pixel precision using motion integration", in *Proc. IEEE PacRim'09*, 2009. *Cited in Sec.* 2.3

[CBS09] Y.-M. Chen, I. V. Bajić, and P. Saeedi, "Coarse-to-fine moving region segmentation in compressed video", in *Proc. IEEE WIAMIS'09*, 2009. *Cited in Sec.* 2.3

[CC00] L. d. F. D. Costa and R. M. Cesar, Jr., *Shape Analysis and Classification: Theory and Practice*, CRC Press, Inc., Boca Raton, FL, USA, first edn., 2000. *Cited in Sec.* 2.1.1

[CCPP14] M. Calemme, M. Cagnazzo, and B. Pesquet-Popescu, "Lossless contour coding using elastic curves in multiview video plus depth". *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014. *Cited in Sec.* 2.4, 3.2, 3.3.1, 3.4, 4.5

[CCPP15a] ———, "Contour-based depth coding: a subjective quality assessment study", in *Proceedings of 2015 IEEE International Symposium on Multimedia (ISM)*, 2015. *Cited in Sec.* 4.5

[CCPP15b] ———, "Depth coding and perceived quality for 3d video", in *Proceedings of QoMEX 2015*, 2015. *Cited in Sec.* 4.5

[CKO+11] G. Cheung, W.-S. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsification", in *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*, pp. 1–6, IEEE, 2011. *Cited in Sec.* 3.1

[Cla85] R. J. Clarke, *Transform Coding of Images*, Academic Press, Inc., Orlando, FL, USA, 1985. *Cited in Sec.* 1.1.1

[CPD13] M. Cagnazzo, B. Pesquet-Popescu, and F. Dufaux, *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*, Chap. 3D Video Representation and Formats, Wiley, 2013. *Cited in Sec.* 3.1

[CPPV07] M. Cagnazzo, S. Parrilli, G. Poggi, and L. Verdoliva, "Costs and advantages of object-based image coding with shape-adaptive wavelet transform". *EURASIP Journal on Image and Video Processing*, vol. 2007, pp. Article ID 78323, 13 pages, 2007, doi:10.1155/2007/78323. *Cited in Sec.* (document), 3.2.3

[CPVZ04] M. Cagnazzo, G. Poggi, L. Verdoliva, and A. Zinicola, "Region-oriented compression of multispectral images by shape-adaptive wavelet transform and SPIHT", in *Proceedings of IEEE International Conference on Image Processing*, vol. 4, pp. 2459–2462, Singapore, Oct. 2004. *Cited in Sec.* 3.2.1, 4.2.1

[CT91] T. M. Cover and J. A. Thomas, *Elements of information theory*, Wiley-Interscience, New York, USA, 1991. *Cited in Sec.* 1.5, 1.5.1

[CW84] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching". *Communications, IEEE Transactions on*, vol. 32 (4), pp. 396–402, 1984. *Cited in Sec.* 1.5.2

[CZM+16] M. Calemme, P. Zanuttigh, S. Milani, M. Cagnazzo, and B. Pesquet-Popescu, "Depth maps coding with elastic contours and 3d surface prediction", in *Proceedings of IEEE International Conference on Image Processing*, 2016. *Cited in Sec.* 3.4

[Dar09] I. Daribo, *Codage et rendu de séquence vidéo 3D; et applications à la télévision tridimensionnelle (TV 3D) et à la télévision à base de rendu de vidéos (FTV)*, Ph.D. thesis, Télécom Paristech, 2009. *Cited in Sec.* 1.3.2

[DCF12] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression", in *Proceedings of IEEE International Conference on Image Processing*, pp. 1541 – 1544, IEEE, Orlando, FL, USA, Sep. 2012. *Cited in Sec.* (document), 2, 2.1.4, 2.2, 2.2.2, 2.3.1, 2.1, 2.3.4, **??**, 2.4, 3.1, 4

[DJC+15] A. Dricot, J. Jung, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Subjective evaluation of Super Multi-View compressed contents on high-end light-field 3D displays". *Elsevier Signal Processing: Image Communication*, p. 15, 2015. *Cited in Sec.* 4

[DM98] I. Dryden and K. Mardia, *Statistical Shape Analysis*, Wiley Series in Probability & Statistics, Wiley, 1998. *Cited in Sec.* 2.1.1

[DPPC13] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, eds., *Emerging technologies for 3D video: content creation, coding, transmission and rendering*, Wiley, 2013. *Cited in Sec.* (document), 4

[DS12] F. De Simone, *Selected Contributions on Multimedia Quality Evaluation*, Ph.D. thesis, EPFL, 2012. *Cited in Sec.* 4.1.2, 4.1.3

[DSGLE11] F. De Simone, L. Goldmann, J.-S. Lee, and T. Ebrahimi, "Towards high efficiency video coding: Subjective evaluation of potential coding technologies". *Journal of Visual Communication and Image Representation*, vol. 22 (8), pp. 734–748, 2011. *Cited in Sec.* 4.1.3

[EG14] M. El Gheche, *Proximal methods for convex minimization of phi-divergences. Application to computer vision.*, Ph.D. thesis, Université Paris-Est Lab. Informatique Gaspard Monge, 2014. *Cited in Sec.* 1.3.1

[EH00] T. Ebrahimi and C. Horne, "Mpeg-4 natural video coding–an overview". *Signal Processing: Image Communication*, vol. 15 (4), pp. 365–385, 2000. *Cited in Sec.* 1.4

[EYUHG10] N. A. El-Yamany, K. Ugur, M. M. Hannuksela, and M. Gabbouj, "Evaluation of depth compression and view synthesis distortions in multiview-video-plus-depth coding systems", in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pp. 1–4, IEEE, 2010. *Cited in Sec.* 3.2.3, 3.3.3, 4.3.4

[Feh04] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV", in *Electronic Imaging 2004*, pp. 93–104, International Society for Optics and Photonics, 2004. *Cited in Sec.* 3, 4.3.4

[FLWM15] E.-C. Forster, T. Lowe, S. Wenger, and M. Magnor, "Rgb-guided depth map compression via compressed sensing and sparse coding", in *Proc. of PCS 2015*, 2015. *Cited in Sec.* 3.1

[FPdW07] D. Farin, R. Peerlings, and P. H. N. de With, "Depth-image representation employing meshes for intermediate-view rendering and coding", in *3DTV-CON 2007 - Capture, Transmission and Display of 3D Video*, 2007. *Cited in Sec.* 3.1

[Fre61] H. Freeman, "On the encoding of arbitrary geometric configurations". *IRE Transactions on Electronic Computers*, vol. EC-10 (2), pp. 260–268, June 1961. *Cited in Sec.* 2.1.2

[Fre78] ———, *Application of the Generalized Chain Coding Scheme to Map Data Processing*, Defense Technical Information Center, 1978. *Cited in Sec.* 2.1.2

[GBG15] M. Georgiev, E. Belyaev, and A. Gotchev, "Depth map compression using color-driven isotropic segmentation and regularised reconstruction", in *Proc. of DCC 20155*, pp. 153–162, 2015. *Cited in Sec.* 3.1

[GG92] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic, Jan. 1992. *Cited in Sec.* 1.1.1

[GLG12] J. Gautier, O. Le Meur, and C. Guillemot, "Efficient depth map compression based on lossless edge coding and diffusion", in *Proceedings of Picture Coding Symposium*, pp. 81–84, Kraków, Poland, May 2012. *Cited in Sec.* (document), 2.3.4, 3.1, 4

[GPT+07] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding: Basics, problems and recent advances". *IEEE Signal Processing Magazine*, pp. 67–76, Sep. 2007. *Cited in Sec.* (document)

[HMK+10] W.-J. Han, J. Min, I.-K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M.-S. Cheon, N. Shlyakhov, K. McCann, T. Davies, and J.-H. Park, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20 (12), pp. 1709 –1720, December 2010. *Cited in Sec.* 1.2.1

[HMS13a] J. Hanca, A. Munteanu, and P. Schelkens, "Lossy contour-coding in segmentation-based intra-depth map coding", in *SPIE Three-Dimensional Image Processing and Applications, 3DIP*, vol. 8650, pp. 865009–865009–8, 2013. *Cited in Sec.* 3.1

[HMS13b] ———, "Segmentation-based intra coding of depth maps using texture information", in *Digital Signal Processing (DSP), 2013 18th International Conference on*, pp. 1–6, July 2013. *Cited in Sec.* 3.1

[HTG08] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment". *Electronics letters*, vol. 44 (13), pp. 9–10, 2008. *Cited in Sec.* 1.1.4

[Huf52] D. Huffman, "A method for the construction of minimum-redundancy codes". vol. 40 (9), pp. 1098–1101, Sep. 1952. *Cited in Sec.* 1.5

[HV92] P. G. Howard and J. S. Vitter, "Practical implementations of arithmetic coding", Tech. Rep., Brown University, 1992. *Cited in Sec.* 1.5.1

[Int05] International Organization for Standardization, Geneva, Switzerland., *ISO 20462: Photography - Psychophysical experimental methods for estimating image quality*, 2005. *Cited in Sec.* 4.1.2

[ISO93] ISO/IEC JTC1 11544, ITU-T Rec. T.82., *Joint Bi-level Image Experts Group: Information technology – progressive lossy-lossless coding of bi-level images.*, 1993. *Cited in Sec.* 3.1

[ISO11] "Call for proposals on 3D video coding technology", Tech. Rep., ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, Mar. 2011, doc. N12036. *Cited in Sec.* (document)

[ITU99] "Subjective video quality assessment methods for multimedia applications", September 1999. *Cited in Sec.* 4.1.2

[ITU04] "Tutorial - objective perceptual assessment of video quality: Full reference television", 2004. *Cited in Sec.* 4.1.3

[ITU07] "Methodology for the subjective assessment of video quality in multimedia applications", January 2007. *Cited in Sec.* 4.1.2

[ITU12a] "General viewing conditions for subjective assessment of quality of sdtv and hdtv television pictures on flat panel displays", August 2012. *Cited in Sec.* 4.1.2, 4.3.3

[ITU12b] "Methodology for the subjective assessment of the quality of television pictures", January 2012. *Cited in Sec.* 4.1.2, 4.1.3, 4.3.1, 4.3.3, 4.4

[Jag11] F. Jager, "Contour-based segmentation and coding for depth map compression", in *VCIP*, 2011. *Cited in Sec.* 3.1

[Jai89] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989. *Cited in Sec.* 1.1.1

[Jay73] N. Jayant, "Adaptive quantization with a one-word memory". *Bell System Technical Journal*, vol. 52 (7), pp. 1119–1144, September 1973. *Cited in Sec.* 1.1.1

[JBB+98] C. L. B. Jordan, S. Bhattacharjee, F. Bossen, F. Jordan, and T. Ebrahimi, "Shape representation and coding of visual objets in multimedia applications — an overview". *Annales Des Télécommunications*, vol. 53 (5), pp. 164–178, 1998. *Cited in Sec.* 1.4.2

[KB92] B. W. Kolpatzik and C. A. Bouman, "Optimized error diffusion based on a human visual model", in *SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology*, pp. 152–164, International Society for Optics and Photonics, 1992. *Cited in Sec.* 1.1.4

[KBE00] J. I. Kim, A. C. Bovik, and B. L. Evans, "Generalized predictive binary shape coding using polygon approximation". *Signal Processing: Image Communication (Elsevier Science)*, vol. 15 (7-8), pp. 643–663, May 2000. *Cited in Sec.* 2.1.1, 3.1

[KGS+08] A. Krutz, A. Glantz, T. Sikora, P. Nunes, and F. Pereira, "Automatic object segmentation algorithms for sprite coding using mpeg-4", in *ELMAR, 2008. 50th International Symposium*, vol. 2, pp. 459–462, Sept 2008. *Cited in Sec.* 1.4.1

[KKM+98] A. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and rate-distortion-based shape-coding techniques". *Proceedings of the IEEE*, vol. 86 (6), pp. 1126–1154, Jun. 1998. *Cited in Sec.* 1.4.2, 2.1.1, 3.1

[Lin] *MPEG4 Segmented Masks*. *Cited in Sec.* 2.3

[LL00] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding". *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 725–743, Aug. 2000. *Cited in Sec.* 3.2.1, 4.2.1

[MÖ07] M. Müller, *Information Retrieval for Music and Motion*, Springer, 2007. *Cited in Sec.* 2.2.1

[MBXV06] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multiview video compression". *Picture Coding Symposium*, vol. 37, pp. 38–39, 2006. *Cited in Sec.* 1.3.3

[MC10] S. Milani and G. Calvagno, "A Depth Image Coder Based on Progressive Silhouettes". *Signal Processing Letters, IEEE*, vol. 17 (8), pp. 711–714, 2010. *Cited in Sec.* 3.1

[MC11a] ———, "3D Video Coding via Motion Compensation of Superpixels", in *Proc. of EUSIPCO 2011*, pp. 1899 – 1903, Aug. 29 – Sep. 2, 2011. *Cited in Sec.* 3.1

[MC11b] ———, "A cognitive approach for effective coding and transmission of 3D video". *ACM Trans. Multimedia Comput. Commun. Appl. (TOMCCAP*, vol. 7S (1), pp. 23:1–23:21, Nov. 2011. *Cited in Sec.* 3.1

[McM97] L. McMillan, *An image based approach to three-dimensional computer graphics*, Ph.D. thesis, University of North Carolina, Chapel Hill, NC, USA, 1997. *Cited in Sec.* 1.3.2

[MFdW07] Y. Morvan, D. Farin, and P. de With, "Depth-image compression based on an r-d optimized quadtree decomposition for the transmission of multiview images", in *Proceedings of IEEE International Conference on Image Processing*, 2007. *Cited in Sec.* 3.1

[MJ99] K. V. Mardia and P. E. Jupp, *Directional Statistics*, Wiley, 1999. *Cited in Sec.* 2.2.2

[MJCPP13a] E. G. Mora, J. Jung, M. Cagnazzo, and B. Pesquet-Popescu, "Depth video coding based on intra mode inheritance from texture". *APSIPA Transactions on Signal and Information Processing*, Jan. 2013, submitted. *Cited in Sec.* 1.3.3

[MJCPP13b] ———, "Initialization, limitation and predictive coding of the depth and texture quadtree in 3d-hevc video coding". *IEEE Transactions on Circuits and Systems for Video Technology*, Feb. 2013, submitted. *Cited in Sec.* 1.3.3

[MJCPP13c] ———, "Modification of the merge candidate list for dependent views in 3D-HEVC", in *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, 2013. *Cited in Sec.* 1.3.3

[MJPPC13] E. G. Mora, J. Jung, B. Pesquet-Popescu, and M. Cagnazzo, "Modification of the disparity vector derivation process in 3d-hevc", in *IEEE Workshop on Multimedia Signal Processing*, vol. 1, Cagliari, Italy, September 2013. *Cited in Sec.* 1.3.3

[MMRH15] M. Maceira, J. R. Morros, and J. Ruiz-Hidalgo, "Region-based depth map coding using a 3d scene representation", in *ICASSP*, 2015. *Cited in Sec.* 3.1

[MMS+08] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. H. N. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality", in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, pp. 245–248, May 2008. *Cited in Sec.* 3.1

[MMS+09] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, T. Wiegand, *et al.*, "The effects of multiview depth video compression on multiview rendering". *Signal Processing: Image Communication (Elsevier Science)*, vol. 24 (1), pp. 73–88, 2009. *Cited in Sec.* (document)

[MPP08] T. Maugey and B. Pesquet-Popescu, "Side information estimation and new schemes for multiview distributed video coding". *Elsevier Journal of Visual Communication and Image Representation*, vol. 19 (8), pp. 589–599, Dec. 2008. *Cited in Sec.* 2.3.4

[MSD+08] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View synthesis for advanced 3d video systems". *EURASIP Journal on Image and Video Processing*, 2008. *Cited in Sec.* 1.3.2

[MSH+13] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, "An overview of tiles in hevc". *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7 (6), pp. 969–977, Dec 2013. *Cited in Sec.* 1.2.1

[MSJ07] W. Mio, A. Srivastava, and S. Joshi, "On shape of plane elastic curves". *International Journal of Computer Vision*, vol. 73 (3), pp. 307–324, 2007. *Cited in Sec.* 2.1.3

[MSMW07a] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17 (11), pp. 1461–1473, Nov. 2007, invited Paper. *Cited in Sec.* 1.3.3

[MSMW07b] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding", in *IEEE International Conference on Image Processing*, vol. 1, pp. 201–204, Oct. 2007. *Cited in Sec.* (document)

[MSW03] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13 (7), pp. 620–636, Jul. 2003. *Cited in Sec.* 1.5.3

[MTZ13] R. Mathew, D. Taubman, and P. Zanuttigh, "Scalable coding of depth maps with r-d optimized embedding". *Image Processing, IEEE Transactions on*, vol. 22 (5), pp. 1982–1995, 2013. *Cited in Sec.* 3.1

[MV93] T. Mitsa and K. L. Varkur, "Evaluation of contrast sensitivity functions for the formulation of quality measures incorporated in halftoning algorithms", in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 5, pp. 301–304, IEEE, 1993. *Cited in Sec.* 1.1.4, 4.4

[MWS06] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications". *IEEE Communications Magazine*, pp. 134–143, Aug. 2006. *Cited in Sec.* (document), 1.5.3

[MZZF11] S. Milani, P. Zanuttigh, M. Zamarin, and S. Forchhammer, "Efficient depth map compression exploiting segmented color data", in *Proc. of IEEE ICME 2011*, pp. 1–6, 2011. *Cited in Sec.* 3.1

[O'C97] K. J. O'Connell, "Object-adaptive vertex-based shape coding method". *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7 (1), pp. 251–255, 1997. *Cited in Sec.* 2.1.1

[OSS+12] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards - including high efficiency video coding (HEVC)". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22 (12), pp. 1669–1684, 2012. *Cited in Sec.* 1.2

[OYVH09] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, "Depth reconstruction filter and down/up sampling for depth coding in 3-d video". *IEEE Signal Processing Letters*, vol. 16 (9), pp. 747–750, Sep. 2009. *Cited in Sec.* 1.3.3, 3.1

[PMC+16] A. Purica, E. G. Mora, M. Cagnazzo, B. Ionescu, and B. Pesquet-Popescu, "Multiview plus depth video coding with temporal prediction view synthesis". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26 (2), pp. 360 – 374, February 2016. *Cited in Sec.* 1.3.3

[PMCPP13] G. Petrazzuoli, T. Maugey, M. Cagnazzo, and B. Pesquet-Popescu, "A distributed video coding system for multi-view video plus depth", in *Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 699–703, Pacific Groove, CA, November 2013. *Cited in Sec.* 1.3.3

[PPCD14] B. Pesquet-Popescu, M. Cagnazzo, and F. Dufaux, eds., *Motion Estimation Techniques*, Elsevier E-Reference Signal Processing, 2014. *Cited in Sec.* 1.1.2

[Ric04] I. E. G. Richardson, *Video Formats and Quality*, pp. 9–25, John Wiley & Sons, Ltd, 2004. *Cited in Sec.* 1.1.4, 4.1

[RS08] D. M. Rouse and S. S., "Understanding and simplifying the structural similarity metric", in *Proceedings of IEEE International Conference on Image Processing*, 2008. *Cited in Sec.* 1.1.4

[Sai04] A. Said, "Introduction to arithmetic coding-theory and practice", Tech. Rep., Hewlett Packard Laboratories, 2004. *Cited in Sec.* 1.5.1, 2.2.3

[Say12] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers Inc., Boston, fourth edition edn., 2012. *Cited in Sec.* 1.1.1, 1.5.2

[SB06] H. R. Sheikh and A. C. Bovik, "Image information and visual quality". *Image Processing, IEEE Transactions on*, vol. 15 (2), pp. 430–444, 2006. *Cited in Sec.* 1.1.4, 4.4

[SBB11] H. Schwarz, C. Bartnik, and S. Bosse, "Description of 3d video technology proposal by fraunhofer hhi. iso", Tech. Rep., IEC JTC1/SC29/WG11 MPEG2011, 2011. *Cited in Sec.* 3.1

[SBdV05] H. R. Sheikh, A. C. Bovik, and G. de Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics". *IEEE Transactions on Image Processing*, vol. 14 (12), pp. 2117–2128, Dec 2005. *Cited in Sec.* 1.1.4

[SC67] G. Snedegor and W. G. Cochran, *Statistical methods*, no. 6th ed, Ames: Iowa State University Press, 1967. *Cited in Sec.* 4.1.1

[Sha48] C. E. Shannon, "A mathematical theory of communication". *The Bell System Technical Journal*, vol. 27, pp. 379–423,623–656, Oct. 1948. *Cited in Sec.* 1.5

[SKJJ10] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, "Shape analysis of elastic curves in euclidean spaces". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33 (7), pp. 1415–1428, Sep. 2010. *Cited in Sec.* (document), 2, 2.1.3

[SKN+10] G. Shen, W.-S. Kim, S. K. Narang, A. Ortega, J. Lee, and H. C. Wey, "Edge-adaptive transforms for efficient depth map coding", in *Proceedings of Picture Coding Symposium*, 2010. *Cited in Sec.* (document), 3.1, 4

[Sma12] C. Small, *The Statistical Theory of Shape*, Springer Series in Statistics, Springer New York, 2012. *Cited in Sec.* 2.1.1

[SOHW12] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22 (12), pp. 1649–1668, 2012. *Cited in Sec.* 1.2, 1.2.1, 1.5.3, 4

[SP96] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6 (3), pp. 243–250, Jun. 1996. *Cited in Sec.* 3.2.1

[SR00] M.-T. Sun and A. R. Reibman, *Compressed Video over Networks*, Marcel Dekker, Inc., New York, NY, USA, first edn., 2000. *Cited in Sec.* 1.1.1

[SSB14] V. Sze, G. J. Sullivan, and M. Budagavi, *High Efficiency Video Coding (HEVC) - Algorithms and architectures*, Springer, 2014. *Cited in Sec.* 1.2

[SSO09] A. Sanchez, G. Shen, and A. Ortega, "Edge-preserving depth-map coding using graph-based wavelets", in *Proc. of* 43^{rd} *Asilomar Conference 2009*, pp. 578 –582, Pacific Grove, CA, USA, Nov. 2009. *Cited in Sec.* 3.1

[Ste46] S. Stevens, "On the theory of scales of measurement". *Science*, (103), pp. 677–680, 1946. *Cited in Sec.* 4.1.2

[TSA14] I. Tabus, I. Schiopu, and J. Astola, "Context coding of depth map images under the piecewise-constant image model representation". *IEEE Transactions on Image Processing*, to appear, 2014. *Cited in Sec.* 3.1

[VCG+12] G. Valenzise, G. Cheung, R. Galvao, M. Cagnazzo, B. Pesquet-Popescu, and A. Ortega, "Motion prediction of depth video for depth-image-based rendering using don't care regions", in *Proceedings of Picture Coding Symposium*, pp. 93–96, Krakow, Poland, May 2012. *Cited in Sec.* 1.3.3

[VM13] A. Vetro and K. Müller, *Depth-Based 3D Video Formats and Coding Technology*, pp. 139–161, John Wiley & Sons, Ltd, 2013. *Cited in Sec.* 1.3, 1.3.3

[VS01] A. Vetro and H. Sun, "An overview of mpeg-4 object-based encoding algorithms", in *Information Technology: Coding and Computing, 2001. Proceedings. International Conference on*, pp. 366–369, IEEE, 2001. *Cited in Sec.* 1.4

[VWS11] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard". *Proceedings of the IEEE*, vol. 99 (4), pp. 626–642, Apr. 2011, invited Paper. *Cited in Sec.* (document), 1.3.3

[WBSS04] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity". *IEEE Transactions on Image Processing*, vol. 13 (4), pp. 600–612, 2004. *Cited in Sec.* 1.1.4, 4.4

[Win05] S. Winkler, *Digital Video Quality: Vision Models and Metrics*, Wiley, 2005. *Cited in Sec.* 4.1.1

[WSW12] M. Winken, H. Schwarz, and T. Wiegand, "Motion vector inheritance for high efficiency 3d video plus depth coding", in *Picture Coding Symposium (PCS), 2012*, pp. 53–56, IEEE, 2012. *Cited in Sec.* 1.3.3

[YO95] Y. Yoo and A. Ortega, "Adaptive quantization without side information using scalar-vector quantization and trellis coded quantization", in *Signals, Systems and Computers, 1995. 1995 Conference Record of the Twenty-Ninth Asilomar Conference on*, vol. 2, pp. 1398–1402 vol.2, Oct 1995. *Cited in Sec.* 1.1.1

[YV09] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding". *Signal Processing: Image Communication*, vol. 24 (1), pp. 89–100, 2009. *Cited in Sec.* 1.3.3

[ZC09] P. Zanuttigh and G. M. Cortelazzo, "Compression of depth information for 3d rendering", in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009*, pp. 1–4, IEEE, 2009. *Cited in Sec.* (document), 3.1, 3.3.1, 3.3.2, 3.3.3, 3.9

[ZJZ+09] B. Zhu, G. Jiang, Y. Zhang, Z. Peng, and M. Yu, "View synthesis oriented depth map coding algorithm", in *Proc. of APCIP 2009*, vol. 2, pp. 104 –107, 2009. *Cited in Sec.* 3.1

[ZL77] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression". *IEEE Transactions on Information Theory*, vol. 23 (3), pp. 337–343, May 1977. *Cited in Sec.* 1.5

[ZRS+12] D. Zhao, Y. Ren, J. Sun, W. Liu, and J. Liu, "Depth map extraction based on geometry", in *Proceedings of IEEE Southeastcon*, 2012. *Cited in Sec.* (document)

[ZZRW13] T. Zhao, K. Zeng, A. Rehman, and Z. Wang, "On the use of SSIM in HEVC", in *Asilomar Conference on Signals, Systems and Computers*, pp. 1107–1111, Ieee, Pacific Grove, California, USA, Nov. 2013. *Cited in Sec.* 1.1.4